A RULE-BASED ALGORITHM FOR THE OPTIMUM CONTROL OF THE GRASP WITH UNEXPECTED DISTURBANCES

By

YUN-GYU KIM

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1993

To

My family

ACKNOWLEDGEMENTS

I would like to thank the Republic of Korea Army for providing me with the financial support during my graduate study in the University of Florida.

Special appreciation and sincere gratitude are due to Professor A. Seireg for his guidance, encouragement, and advice toward this work and for his valuable assistance, discussion, and fruitful hints. Also, I would like to express my respect and thanks to my committee members, Dr. J. Duffy, Dr. C. D. Crane, Dr. S. Smith and Dr. H. W. Doddington.

I am thankful to my wife Young-ye and my children Hyun-Tae and Ji-Hae for their loving patience, understanding, and encouragement throughout my graduate work.

TABLE OF CONTENTS

Abstract of Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A RULE-BASED ALGORITHM
FOR THE OPTIMUM CONTROL OF THE GRASP
WITH UNEXPECTED DISTURBANCES

By

Yun-Gyu Kim

August 1993

Chairperson: Ali A. Seireg
Major Department: Mechanical Engineering

The objective of this study is to develop the algorithms necessary to control the optimum grasp with multifingered end effectors for unexpected changes in the external conditions.

Control algorithms are developed for incorporating in real time the necessary modification of the grasp to insure that slip does not occur due to unexpected changes in the applied forces and moments beyond the allowable limits afforded by the initial grasp.

The primary emphasis is to provide the simplest sensor system and logic scheme with programmed rules to execute the best change of the orientation of the object and the normal forces necessary to prevent any impending slip before it occurs.

A 3-finger hand with frictional forces and moments is used for illustration. Two sensor systems are considered in structuring the real time control of the grasp.

Several computer simulation examples are run to evaluate the feasibility of the developed algorithms and the limitations on their application.

This investigation is intended primarily for application in light weight manipulators handling delicate objects where compliance rather than inertia becomes the dominant factor.

CHAPTER 1
INTRODUCTION

1.1 Introduction

A gripper, hand, or end-effector, is the mechanism placed at the end of a robot arm, enabling it to pick up work pieces and hold, manipulate, transfer, place and release them accurately at a given location. Consequently,the gripper is the most important mechanical interface between the robot and its environment. Without the gripper, in many circumstances, the robot can not function effectively.

In spite of the importance of the gripper, practical gripper technology is relatively under-developed at this time. This is because research and development work on grippers is fragmented and somewhat uncoordinated. The results of academic research may not quickly reach the practicing engineer who is charged with implementing robot design and gripper procurement for particular applications.

The modern robot is a computerized multipurpose machine inherently capable of being reprogrammed for different tasks. It is necessary to develop a control system which can manage the gripper under unexpected disturbance when it grasps an object.

1.2 Review of Literature

The literature related to the design and control of multifingered hands is too extensive to review in a comprehensive manner. The following is a brief description of some of the published work which is relevant to the problem under investigation.

1.2.1 Design for Multi-fingered Hands

The design and development of the grasp by robotic fingers has been a good and important research topic for the past two decades. The book on "Mechanical Hands Illustrated" edited by Sadamoto Kuni [1984] which includes detailed description of large number of hands for a wide variety of application illustrates the considerable interest in this subject. One of the earlier researchers, Crossley [1975], designed a three fingered hand for use on a remotely controlled robot. It was able to pick up a tool and draw it into a nested grip against the palm, and was also able to hold a pistol-grip tool such as an electric drill and pull the trigger.

1.2.2 Modeling Contacts with Soft Fingers

Cutkosky and Wright [1986] used a shearing model to describe the contact friction. They developed models of pointed, curved, flat, soft, and soft-curved fingertips and compared them in terms of their contribution to the stiffness

and stability of a simple grasp. They concluded that there are numerous advantages to using compliant materials for the fingers of a robot gripper. The direct advantages are as follows:

1. Compliant materials distribute contact forces, avoiding damage to the gripped object.

2. Compliant materials have large coefficients of friction, making it possible to use lower griping forces.

3. Compliant griping surfaces have more kinematic coupling at each contact with an object so that fewer contacts are required to completely restrain an object. For example, two soft fingers can always restrain an object in space but two hard fingers can not.

One of the important problems in the articulated robot hands is the control of the initial impact force, which is produced at the onset of contact between the robot finger and the object.

Kahng and Amirouche [1987] presented the derivation of the maximum impact force equation for a general two-body collision problem using the energy principle. The equation of the impact force utilizes the kinetic energy, the strain energy stored in the elastic bodies, and the energy lost due to structural damping. They compared the results with data obtained from an experiment.

Howe, Kao, and Cutkosky [1988] investigated finding the magnitudes of applied moment and force which will cause a robot finger to slip on the surface of a grasped object. They described an improved model which includes torsion-shear interaction. Their experimental measurements suggest that a simple linear function of torsion and shear magnitudes will adequately predict the onset of slip in many tasks.

Sinha and Abel [1989] developed a model which utilizes a contact-stress analysis of an arbitrarily shaped object in a multifingered grasp. The fingers and the object were all treated as elastic bodies and the region of contact was modeled as a deformable surface patch. The relationship between the friction and normal forces was nonlocal and nonlinear in nature and departs from the Coulomb approximation.

The superior performance of human fingers motivated Akella and Cutkosky [1989] to create soft, anthropomorphic fingertips for dexterous manipulation. They presented an attempt to model soft fingertips filled with powders or plastic fluids. They reported sensitivity studies to help in the choice of an appropriate model. The visco-plastic nature of the fingertip affected the dynamics of manipulation by dissipating energy. The results of their computer simulation showed that soft fingertips can help stabilize the grasp and reduce the demands on the control system.

1.2.3 Analysis and Optimization of the Grasp

The analysis and optimization of the grasp is a subject of primary concern for designing mechanical grippers. One of the early investigations related to this area was undertaken by Yuan and Freudenstein [1971] who developed the concept of screw coordinates in terms of motor algebra, and applied it to the kinematics and static equilibrium of grasping rigid bodies. They also developed the laws of the composition and transformation of screw coordinates and their application to this class of problems.

Holtzman and McCarthy [1985] presented a procedure for computing the frictional forces required to restrain an object using a three fingered hand for any given surface normals, the contact points and magnitude of the applied normal forces. They used the screw theory and point contact model with friction which allows the fingertip to be free to rotate about the point of contact.

Fearing [1986] presented a method for stably grasping two dimensional polygonal objects considering a dexterous hand with two fingers. He developed the basic constraints on the vertex angles of the object for a stable grasp using two fingers.

Kerr and Sanger [1987] introduced 6 x 6 stiffness matrices for computing the friction forces and the finger normal forces in contact with any arrangement of point contacts having known frictional properties. They established

the relationship between externally applied loads and contact forces in the presence of limiting friction which can be used for evaluating the resulting bodily motion, with or without slip at the contact. They treated the excess frictional force as a disturbance on the original system, to find a body twist which in turn will give a modified set of local forces. They used the screw theory and point contact model with friction which allows the fingertip to be free to rotate about the point of contact without frictional moment resistance.

Van-Duc Nguyen [1987] presented fast and simple algorithms for directly constructing stable grasps in 3-D by developing a simple geometric relation between the stiffness of the grasp and the spatial configuration of the virtual springs at the contacts. He introduced linear and angular spring models in 3-D to construct stiffness matrices. Their main results show that all 3-D force-closure grasp can be made stable. They also found that the stiffness matrix K of the grasp is the sum of two matrices Ks and Kp (Ks depends on the spatial configuration of the virtual springs. Kp depends on whether the finger stick or slide on the straight edges of the object).

Kumar and Waldron [1987] obtained the multifinger reaction forces and friction forces in the grasp by two suboptimal procedures. In each procedure, a pseudo-inverse technique was used to solve the undetermined linear equations. The purpose of this method is to reduce computational time so

that it can be used in real time control with currently available computer hard-ware. They proposed two approaches. In both cases, they decoupled the sub-problems for finding forces parallel to the x-y plane and then forces parallel to the wrench axis. They also used the screw theory and point contact model with friction which allows the fingertip to be free to rotate about the point of contact.

Badreldin and Seireg [1988,1990] developed a generalized optimization algorithm for minimizing the maximum finger normal forces when grasping a moving object. They considered all frictional forces and moments for any multi-finger end effector when handling objects of any shape along given trajectories in a given time. This technique is particularly useful in designing end effectors for handling delicate objects by planning the number of fingers and the corresponding best locations for the grasp by the different fingers.

Demmel and Lafferiere [1989] described a procedure to compute the grasping forces among the fingers making point contact with an object. This scheme reduces the nonlinear optimization problem to a generalized eigenvalue problem. They assumed that the contact points between fingers and object are fixed and the contacts are hard point contacts with friction(it cannot transmit any torque). For the 2-D case, they concluded that the optimal grasping forces, in the sense of minimizing the dependence on friction, are those for which

the angles and the resultant force have with the corresponding normals are all equal(in absolute value).

Nakamura, Nagai and Yoshikawa [1989] developed a method of evaluating contact stability which is the ability for the end-effector to maintain contact with an object without slipping when the object is subjected to disturbing external dynamic forces.

Yoshikawa and Nagai [1991] proposed a new definition of grasping and manipulating forces for multi-fingered robot hands. The grasping force is defined as an internal force that satisfies the static friction constraint. Then the manipulating force is defined as a fingertip force satisfying the following three conditions:

1. It produces the specified resultant force.
2. It is not in the inverse direction of the grasping force.
3. It is orthogonal to the grasping force component.

They also introduced the concept of a grasping focus to verify grasp mode and to calculate grasping forces. They used the following assumptions:

1. Each fingertip makes a frictional contact with the rigid object.
2. The contact points are not located on a straight line.
3. The mechanism of each finger is such that each fingertip can exert a force on the object in any direction.

1.2.4 Control of the Grip and Manipulation

An important area of study in robotics is the control of the grip by the end-effector. The following is a brief summary of some of the published papers in this area which are relevant to the proposed study.

Raibert and Craig [1981] proposed a hybrid position/force controller that uses a wrist mounted force sensor to control the manipulator trajectories in a task oriented Cartesian coordinate system. The method is a straightforward approach to the problem of controlling forces and positions simultaneously. Although this technique is straightforward and feasible, it requires extensive computations and would therefore be difficult to implement in real time control.

Arimoto, Miyazaki and Kawamura [1987] proposed a cooperative motion control scheme for a set of multiple robot arm or fingers. This scheme is easily implemented in the case of a positioning task and it can be extended to the case of path tracking control, force control, and hybrid position/force control.

Cole, Hauser and Sastry [1988] considered the manipulation of objects of arbitrary shapes by multi-fingered hands. They treated the contact between the object and the fingers as rolling contacts, i.e., the fingertip rolls without slipping on the surface of the object. They also presented a control law to dynamically control the sliding motion of the fingertips along the object surface.

Grupen and Weiss [1991] presented a multiple resolution object representation in the force domain which facilitates grasp synthesis. They also proposed a useful method for describing grasp objectives and a control strategy for designing the geometry of contact mediated by force domain goals. They showed that at a chosen configuration(null space), the grasp forces can increase arbitrarily without producing net forces on the object. The required contact forces can be made to satisfy the friction cone at each contact by squeezing the object.

Payandeh and Goldenberg [1991] presented a model of the manipulator in contact with a rigid environment and proposed a robust control architecture based on a general theory of servomechanism for controlling the contact force. They concluded that the controller can not be made fast by increasing the gain of the servo-compensator without jeopardizing the closed-loop stability of the system. However, the presence of the compliant material between the end-point of the manipulator and the rigid environment introduces a finite stiffness which allows the system to achieve a fast and stable robust response.

1.2.5 Knowledge Based Expert Systems

There has been considerable interest in recent years to incorporate developments in artificial intelligence and knowledge based systems in the control of the grip. The

following is a brief summary of some of the relevant studies.

Randy C. Brost [1988] presented an algorithm for automatic planning of robot grasping motions that are insensitive to bounded uncertainties in the object's location. The algorithm plans parallel-jaw grasping motions for arbitrary two-dimensional polygonal objects, which need not be of uniform density.

Stansfield [1991] proposed a knowledge-based approach to describe a general purpose robotic grasping system for use in unstructured environments. Using computer vision and a compact set of heuristic, the system automatically generates the robot arm and hand motions required for grasping an unmodeled object. He implemented a two-stage model of grasping. Stage one is an orientation of the hand and wrist and a ballistic reach toward the object. Stage two is hand preshaping and adjustment. Visual features are first extracted from the unmodeled object. These features and their relations are used by an expert system to generate a set of valid reach/grasps for the object.

A study which demonstrates the feasibility of utilizing rule based computer systems in regaining stable control after it loses stability due to fast occurring unexpected external effects is reported by El-Deen and Seireg [1987]. They showed that the use of a preprogrammed rule-based system can be used to improve a car's ability to perform safely, even when traveling at high speeds on low-friction surfaces by

corrective steering. They developed a computer control algorithm to implement optimized corrections based on real time sensing of the relevant variables. A simulation of the proposed real time computer control system showed that it could considerably enhance the vehicle stability and ability to maneuver safely at relatively high speed on icy roads.

1.2.6 Sensing

Manipulator task categories and motion phases require various hand-based information systems to meet the control performance requirements. One of the early investigations in this area was undertaken by Bejczy [1977]. He presented the effect of proximity, tactile and force/torque sensors on the performance of remote manipulator control. He also presented an over view on various experimental hand-based information systems which provide the manipulator controller with some non-visual "awareness" of the task environment. He concluded that the use of proximity sensor information can considerably improve the performance of remote manipulator control. He suggested several means for improving the performance of manipulator control. They were as follows:

1. Integrated display of visual and proximity sensor information.

2. Computer-aided use of proximity sensor information for control.

3. Adding tactile information to the proximity information.

4. Introducing more dexterous terminal devices.

Dario and Buttazzo [1987] presented the design, implementation, and testing of an artificial sensing system incorporating an articulated robot finger. Primary aim in their work was to set up the hardware and software tools necessary for investigating basic issues in artificial tactile perception.

Yim and Seireg [1988] demonstrated that a computer-controlled optical system provides a relatively simple and computationally fast technique for cooperative motion identification and tracking of objects. They developed an effective algorithm for extracting the position and orientation information of the scan line on a target, with specially coded grid lines, placed on the surface. It can be used for performing and assembly or disassembly operations of moving objects in manufacturing and space station applications.

1.3 Objective of the Study

The objective of this study is to develop a preprogrammed knowledge-based system for adjusting the preplanned grip in order to avoid loss of grasp or slip due to unexpected external or internal conditions in the end effector's

environment. The study includes the following :

1. Developing easily implementable control algorithms for the optimum grip based on the number of fingers and the geometry of the object to sustain the dynamic loads which are to be expected during the handling task.

2. Developing a strategy for sensing any unexpected changes by a comprehensive sensor system which can monitor the direction and magnitude of frictional force and normal force at the fingertip or by a sensor system with normal force transducers only at the fingertips.

3. Using the sensing information for the identification of any unexpected disturbances as they occur.

4. Developing preprogrammed control algorithms for the best sequence of actions necessary to avoid impending slip. These may include changes in the normal forces applied by the fingers as well as rotations of the object according to the preprogrammed algorithm.

5. Structuring a coordinated control strategy to incorporate the knowledge base and sensor information for real time control. The program would predict, initiate control actions, verify the predictions and readjust the actions at every time increment after the sensors indicate that the allowable tolerance limit is violated.

6. A computer simulation to investigate the feasibility of the proposed control concept and any limitations on its implementation.

7. The simulation will be based on a three finger hand configuration for handling a cubical object as an illustration for proving the concept. Figure 1.1 is a flow diagram illustrating the outline of the control strategy.

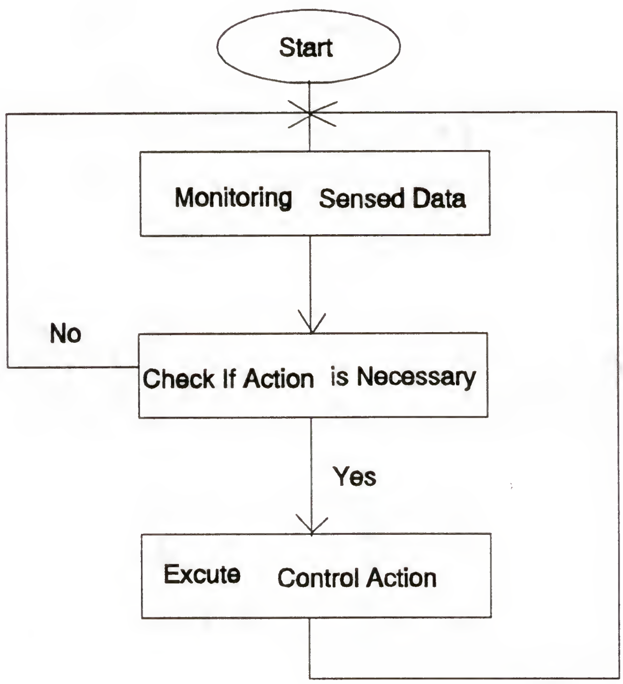

Figure 1.1 : Flow Diagram of the Control Strategy

CHAPTER 2
ANALYSIS OF SOFT CONTACT

2.1 Ideal Independent Springs

2.1.1 Linear Spring Model

A linear spring i can be characterized by its stiffness constant K_i, and its direction which is assumed to be oriented along the normal to the surface at the contact point as shown in Figure 2.1. A linear spring exerts a pure force(i.,e., no moment) on the object if and only if it is applied at a point and it's line of action is along the normal direction. Rotations about the contact point have no effect on this type of spring.

2.1.2 Angular Spring Model

A soft finger contact can resist rotations of the object about the axis normal to the surface at the point of contact (Figure 2.1). The considered angular spring K_i exerts a pure moment if and only if the object is rotated about the normal axis without translation.

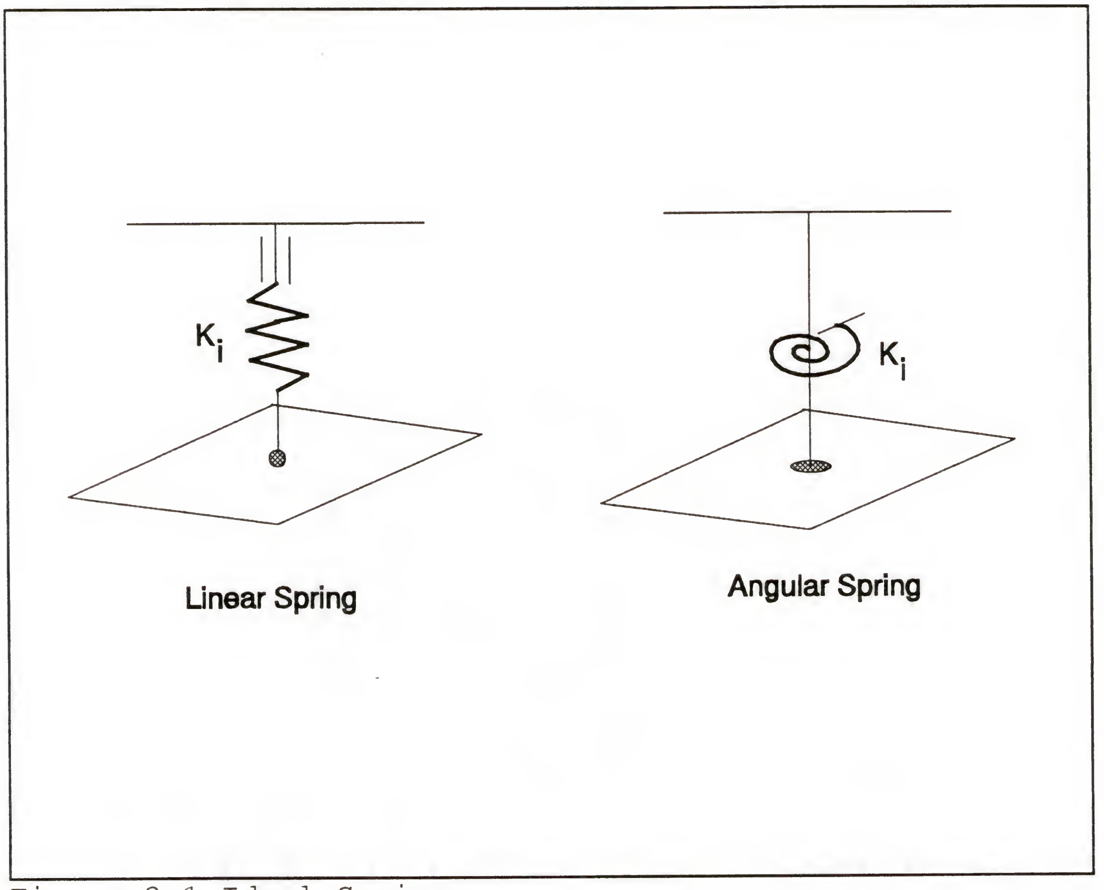

Figure 2.1 Ideal Springs

2.2 Stiffness of the Virtual Spring

The interface between the fingertip and the object can be modelled as a system of linear and angular springs attached at the contact point as in Figure 2.2. The stiffness of the i_{th} fingertip in the contact can be represented by a matrix and relative to a global reference frame, the stiffness is

$$K_{ci} = \begin{bmatrix} & & & \kappa_1 & 0 & 0 \\ & 0 & & 0 & \kappa_2 & 0 \\ & & & 0 & 0 & \kappa_3 \\ \kappa_4 & 0 & 0 & & & \\ 0 & \kappa_5 & 0 & & 0 & \\ 0 & 0 & \kappa_6 & & & \end{bmatrix}_i \tag{2.1}$$

$$K_i = T_i \, K_{ci} \, T_i^{-1} \; . \tag{2.2}$$

where T_i is a general screw transformation matrix of i_{th} contact given by following equation[1].

$$T_i = \begin{bmatrix} \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} & 0 \\ \begin{bmatrix} l_z y - l_y z & m_z y - m_y z & n_z y - n_y z \\ l_x z - l_z x & m_x z - m_z x & n_x z - n_z x \\ l_y x - l_x y & m_y - m_x y & n_y x - n_x y \end{bmatrix} & \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix} \end{bmatrix} \tag{2.3}$$

The stiffness of the entire object is the sum of the stiffness matrices

$$K = \sum_{i=1}^{n} K_i . \tag{2.4}$$

If the global displacement(twist) is known, the external force(wrench) is

---

1. Appendix A

$$W_e = K \cdot t \tag{2.5}$$

where t is the twist(displacement vector).

$$t = (\delta x \ \delta y \ \delta z \ \delta\theta_x \ \delta\theta_y \ \delta\theta_z)^t \tag{2.6}$$

If the stiffness matrix K is invertible, the displacement of the object is

$$t = K^{-1}W_e. \tag{2.7}$$

and with respect to the contact frame, the displacement is

$$t_i = T_i K^{-1} W_e. \tag{2.8}$$

Where We is the given external force(wrench) including the inertia force.

$$We = (\delta f_x \ \delta f_y \ \delta f_z \ \delta m_x \ \delta m_y \ \delta m_z)^t \tag{2.9}$$

2.3 Relationship Between Moment and Force

A common assumption in grasping analysis is that the friction limits due to torsion and shear are independent, so that the onset of slipping in rotation doesn't depend on applied tangential load(frictional force), and linear slipping doesn't depend on applied moment. However, Howe, Kao and

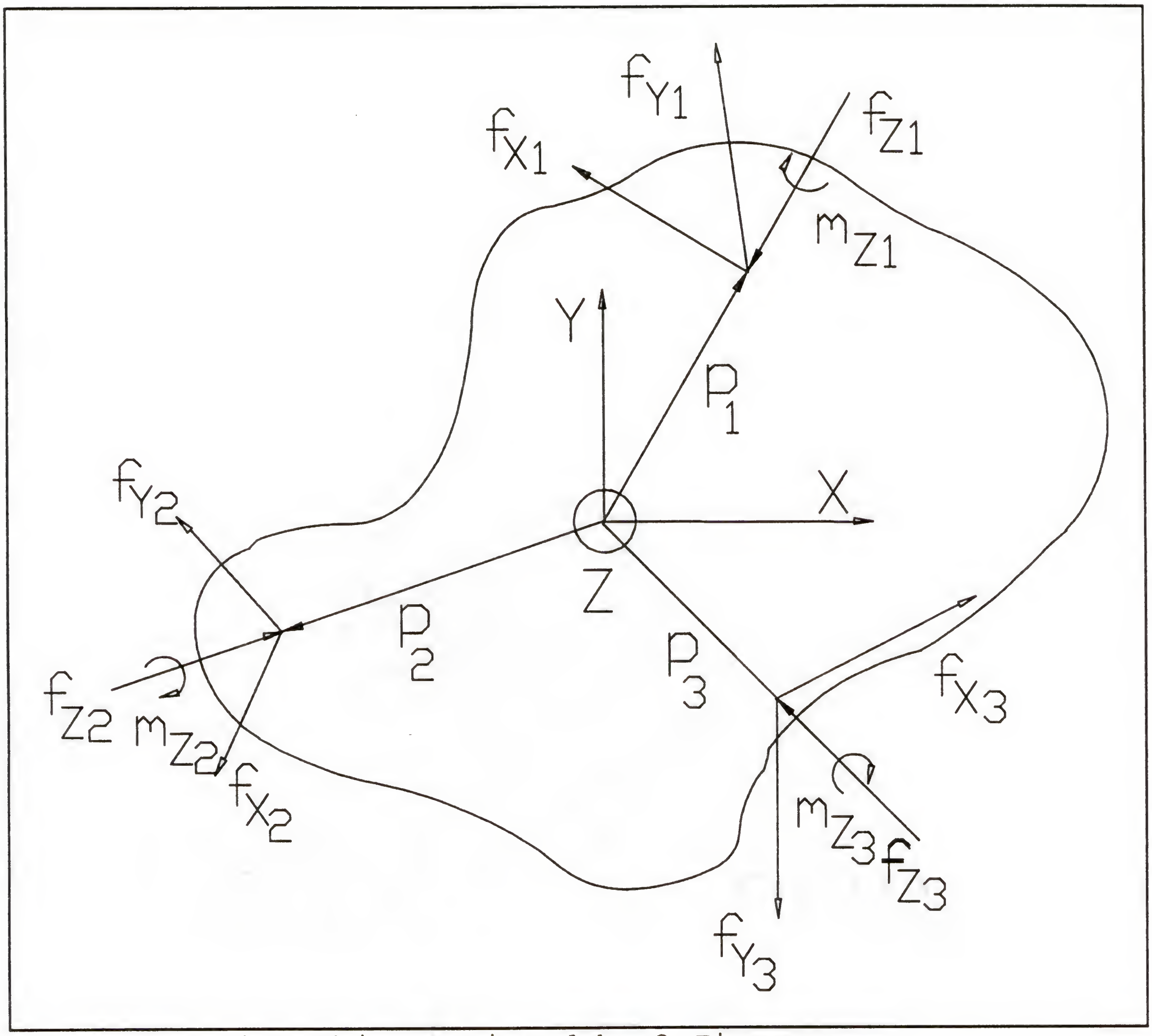

Figure 2.2 3-D Object Gripped by 3 Fingers

Cutkosky [1988] measured on set of slipping as a function of applied moment and tangential force at several different normal forces. They suggested that if the prevention of the slip is of paramount concern in a manipulation task, then a straight line drawn between the maximum moment and maximum shear is a simple, somewhat conservative estimate of the slipping limit. A linear relationship between normal force

and moment was used by Badreldin and Seireg [1988,1990] in order to minimize the maximum finger normal force by a linear programming method.

Both of the above papers proposed a simple constraint equation.

$$\mu \mid F_n \mid \geq f_t + A \mid M_n \mid \tag{2.10}$$

where f_t is the tangential frictional force on the contact,

$$f_t = \sqrt{(f_x^2 + f_y^2)} \tag{2.11}$$

μ is the coefficient of friction, F_n is the normal force, M_n is the moment, and A is a proportionality constant between the torsion and shear limits. If the relationship between the normal force and contact area is known, the constant A can be derived as follows:

$$M = \int dM = \int_0^R 2 \cdot \pi r \cdot dr \cdot \mu \cdot \frac{F_n{}'}{S} \cdot r = \frac{2}{3} \cdot \pi \cdot R \cdot \mu \cdot F_n{}' \tag{2.12}$$

$$A = \frac{3}{2 \cdot \pi \cdot R} \tag{2.13}$$

where S is the area of the contact at the fingertip, and R is the radius of the area.

CHAPTER 3
OPTIMIZATION OF THE GRASP

3.1 Introduction

The optimization problem is to find the optimal gripping configuration(i.e., the location of fingers on the surface of the object) which generates the minimum max normal force between fingers and object while moving it along a predetermined path. For a particular grasp configuration, the maximum force of the forces necessary to hold the object and move it along a predetermined path without slip can be obtained by linear programming. The problem is solved in three steps.

The first step is to find the ratios between the finger forces, with the smallest possible maximum finger normal force, which does not produce any resultant wrench or twist of the grasped object. This ratio should satisfy the frictional constraints.

The second step is to determine the smallest incremental finger normal forces which are required to resist all the external forces including weight and inertia forces.

In the third step, the solutions from the first and second steps are combined to determine the minimum multiplier

for the finger gripping force ratio which is necessary to grasp the object without slipping at all contact areas.

3.2 The Static Gripping Ratio

In this step, the object is assumed to be weightless. Its equilibrium is defined by six linear equations which define the relationship between the finger forces for any specific gripping configurations. These forces include the normal forces applied by the fingers as well as all the possible frictional forces and moments at the contact areas between the fingers and the object.

In the general case, the equilibrium equations are statically indeterminate and linear programming is used to obtain a solution. The linear objective function is to minimize the maximum frictional components of finger forces. It insures the grasp stability by decreasing its dependency of the frictional components. The objective function and constraints equations are as follows.

$$Min(U) = f_{\max} + A \cdot M_{\max} \tag{3.1}$$

subject to

$$\sum_{i=1}^{n} T_i W_i = 0 \tag{3.2}$$

$$|f_{x_i}| \leq f_{\max} \tag{3.3}$$

$$|f_{y_i}| \leq f_{\max} \tag{3.4}$$

$$|m_{z_i}| \leq m_{\max} \tag{3.5}$$

$$-\sum_{i=1}^{n} f_{z_i} \leq -1 \tag{3.6}$$

Where f_{zi} , f_{xi} and f_{yi} are the forces which do not disturb object's equilibrium, f_{zi} is the normal force f_{xi} and f_{yi} are perpendicular frictional forces and m_z is frictional moment at i_{th} contact point. W_i is the local wrench due to the global twist, and A is the proportionality constant between the normal force and moment.

$$W_i = (\delta f_x \ \delta f_y \ \delta f_z \ \delta m_x \ \delta m_y \ \delta m_z)^t{}_i \tag{3.7}$$

The solution of the gripping forces and moments is then checked by frictional constraint equation to insure that no slip occurs during the initial grasp.

$$\mu \cdot f_{z_i} \geq \left[\sqrt{(f_{x_i})^2 + (f_{y_i})^2)} + |A \cdot M_{z_i}|\right] \tag{3.8}$$

If a solution does not exist, or if the frictional constraint equations are violated, the grasp is not attainable at this specific grasping configuration and the finger contact locations should be modified.

When a solution is obtained and is found to satisfy the frictional constraints the 1st grasping ratio is found by normalizing the obtained solution as:

$$(F_{zi})^1 = \frac{1}{f_{zmax}} \cdot (F_{zi}) \tag{3.9}$$

where

$$f_{zmax} = \max(f_{zi}\ ,\ i = 1,\ n) \tag{3.10}$$

If any of the finger normal forces is zero in the 1st grasping ratio, another grasping ratio is to be evaluated in which all fingers are constrained to be active. The last constraint equation is to be substituted by:

$$-f_{z_i} \leq -1 \quad i = 1,\ n. \tag{3.11}$$

If a solution is obtained, it is then normalized as:

$$(F_{zi})_n = \frac{1}{f_{zmax}} \cdot (F_{si}) \quad i = 1,n \tag{3.12}$$

If a solution does not exist, then the only possible gripping ratio is $(F_{zi})^1$ $i = 1,\ n$. A scala multiplier is then evaluated for each finger such that when multiplied by the finger normal force it would provide the required frictional forces and moment.

$$\beta_{si} \geq \frac{1}{\mu_i \ \beta \cdot f_{zi}} \cdot \left[\sqrt{f_{x_i}{}^2 + f_{y_i}{}^2} + |A \cdot m_{zi}|\right] \tag{3.13}$$

The max multiplier is found as:

$$\beta_{smax} = \max(\beta_{si}, i = 1, n) \tag{3.14}$$

The second gripping ratio is evaluated using the following equation:

$$(F_{zi})^2 = (F_{zi})^1 + \frac{1}{\beta_{max}} [\ (F_{zi})_n - (F_{zi})^1\ ] \quad i = 1,\ n \tag{3.15}$$

which describes the gripping ratio as a linear combination of the 1st gripping ratio and the 2nd normalized solution.

The dynamic incremental finger forces at the contacts are evaluated by considering all the locations along the prescribed trajectory. The resultant force due to inertia, weight, and external force and moment vector, at the considered instants of time is transformed in the object frame.

3.3 Evaluation of the Frictional Forces and Moments

The frictional forces and moments terms of a griping system can be decoupled by assuming a virtual spring system which can resist forces only along the normal axis at the contact points. In this step, the required minimum frictional forces and moments are obtained without considering the

frictional constraints by linear programming. The frictional constraints can be adjusted in next step. The objective function which was used in the previous step is also used this step. This objective function enhances the grasp stability by minimizing the max frictional components necessary to support the grasped object. Constraints in this step are as follows.

$$\sum_{i=1}^{n} \left(T_i \cdot K_i \cdot t_i + T_i \cdot W_{r_i} \right) = W_e \tag{3.16}$$

$$\left| f_{r_{xi}} \right| \leq f_{r_{max}} \tag{3.17}$$

$$f_{r_{zi}} \geq 0 \tag{3.18}$$

$$\left| f_{r_{yi}} \right| \leq f_{r_{max}} \tag{3.19}$$

$$\left| m_{r_{zi}} \right| \leq m_{r_{max}} \tag{3.20}$$

Where T_i, K_i are the general screw transformation matrix and the stiffness matrix of i_{th} contact respectively, and t_i , W_{r_i} are the local twist and wrench due to the external wrench W_e respectively. The force components f_{r_x}, f_{r_y}, and f_{r_z} are the forces which do not disturb object's equilibrium where f_{r_z} is normal force f_{r_x} and f_{r_y} are perpendicular frictional forces and m_{r_z} is frictional moment at i_{th} contact point.

3.4 Determination of the Dynamic Multipliers

At this stage of the procedure, two minimum multipliers for the gripping ratios(if both exist) can be evaluated to insure that no slip or zero pressure occur between the fingers and the object. These are obtained by solving the following frictional constraint equations.

$$(\beta_i^k f_{z_i}^{\ k}) + f_{r_{x_i}} \le \frac{1}{\mu_i}[((\beta_i^k f_{x_i}^{\ k} + f_{r_{x_i}})^2 + (\beta_i^k f_{y_i}^{\ k} + f_{r_{y_i}})^2)^{0.5} + A\left|\beta_i^k m_{z_i}^{\ k} + m_{r_{z_i}}\right|] \tag{3.21}$$

The maximum of each of the two sets of multipliers are found as:

$$\beta_{max}^{\ k} = \max(\beta_i^k, i = 1, n \ k=1,2) \tag{3.22}$$

The actual grasping forces according to each gripping ratio are therefore calculated from:

$$F_{g_i}^{\ k} = \beta_{max}^{\ k} \cdot (f_i)^k \quad 1=1,n \quad k=1,2 \tag{3.23}$$

The total generated forces along the predetermined path are evaluated using the following equation.

$$(F_{c_i}^{\ k}) = (F_{g_i})^k + (F_{d_i})^k \quad i=1,n \quad k=1,2 \tag{3.24}$$

The max finger normal force that is generated along the predetermined path when any of the two grasping ratios are used to move the object is determined from:

$$N_{\max}{}^{k} = \max(f_{gi}{}^{k}\ ,\ f_{ci}{}^{k}) \tag{3.25}$$

where $f_{g_i}{}^{k}$ and $f_{c_i}{}^{k}$ are the normal components of $F_{g_i}{}^{k}$ and $F_{c_i}{}^{k}$. The lowest of the two maximum normal forces generated along the entire trajectory is determined from:

$$N_{\max} = \min(N_{\max}{}^{k}\ , k=1,2) \tag{3.26}$$

$N_{\max}$ determines which set of forces are used at the considered gripping configuration. The grasp ratio to be used is the one that generates $N_{\max}$ along the trajectory.

The Figure 3.1.1 and 3.1.2 are illustrations of how the combined results of step1 and step2 are used, when the frictional coefficient is assumed to be 0.1.

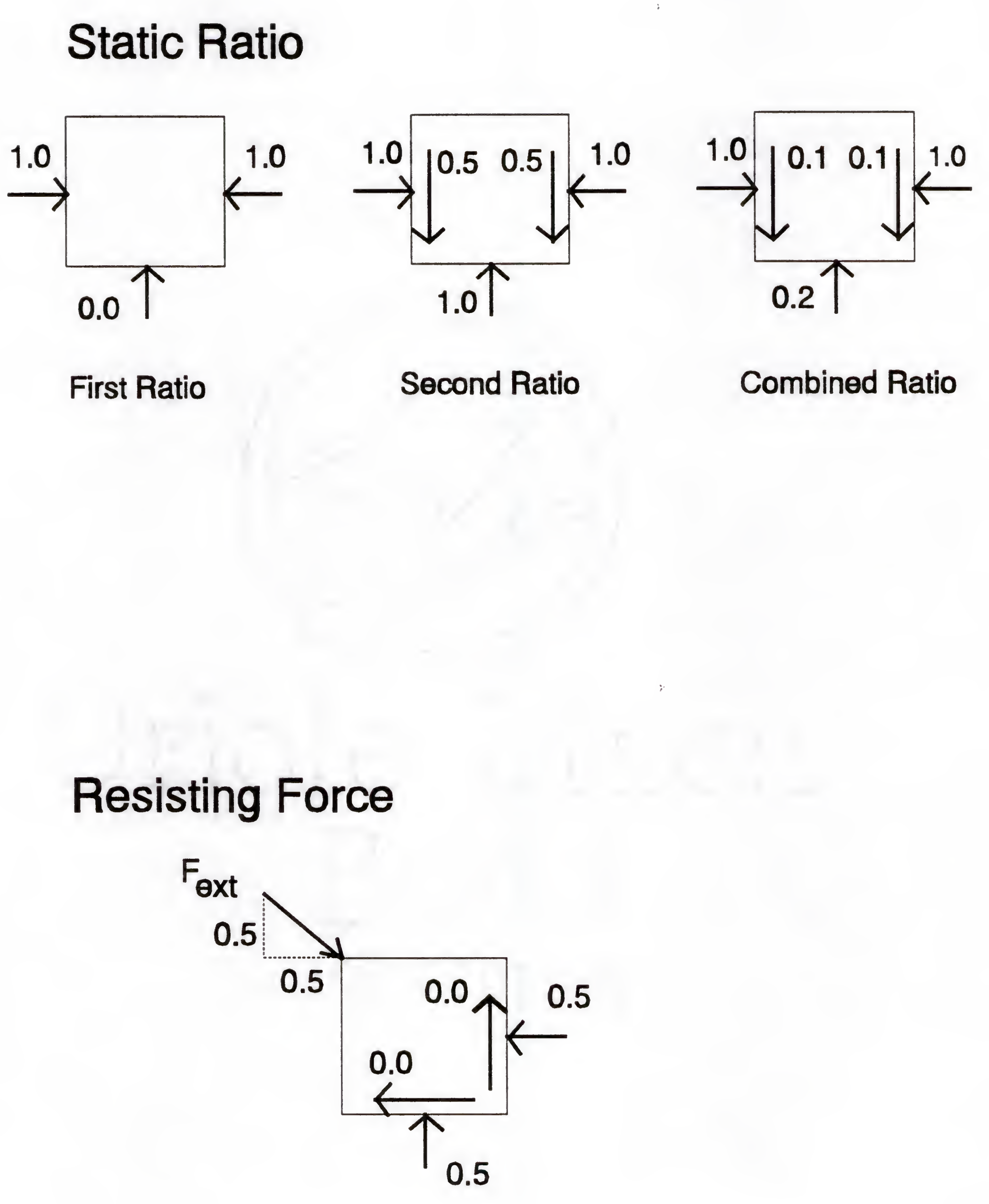

Figure 3.1.1 Static Ratio and Resisting Forces

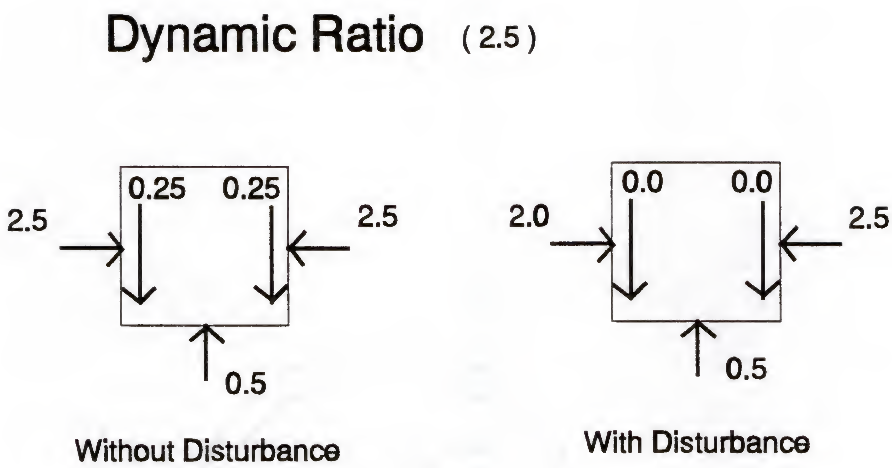

Figure 3.1.2 Dynamic Ratio

3.5 Flow Charts of the Optimization

Several computer programs in c language were developed to find the optimal configuration of three fingered gripper when grasping a cube or a sphere for any given movement. A test program was also developed to confirm the results of the optimization programs. The optimization algorithm follows the same approach as the one developed by Badreldin and Seireg [1988,1990].

Flow charts for the total analysis and optimization procedure are given in Figure 3.2 to 3.6.

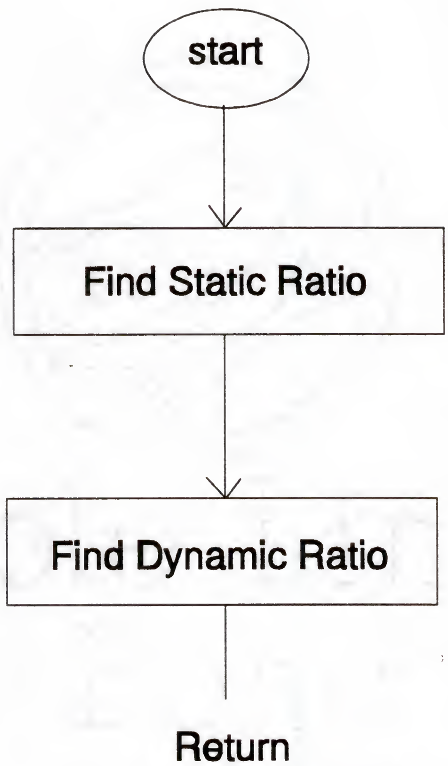

Figure 3.2 Basic Function Flow

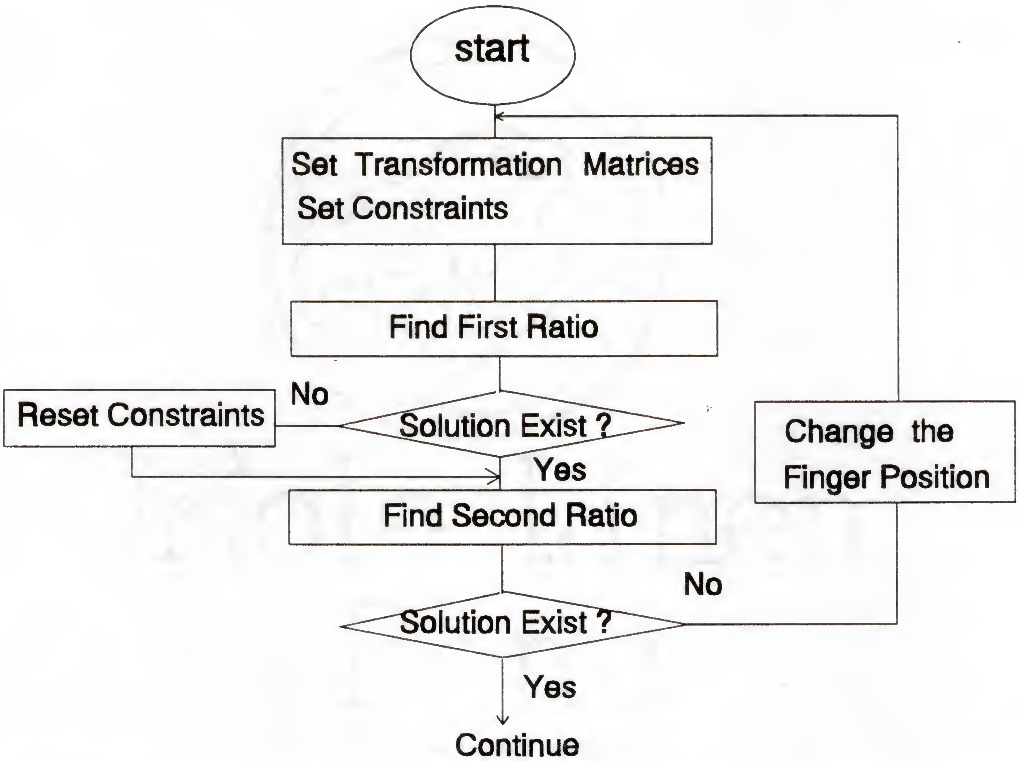

Figure 3.3 Static Ratio

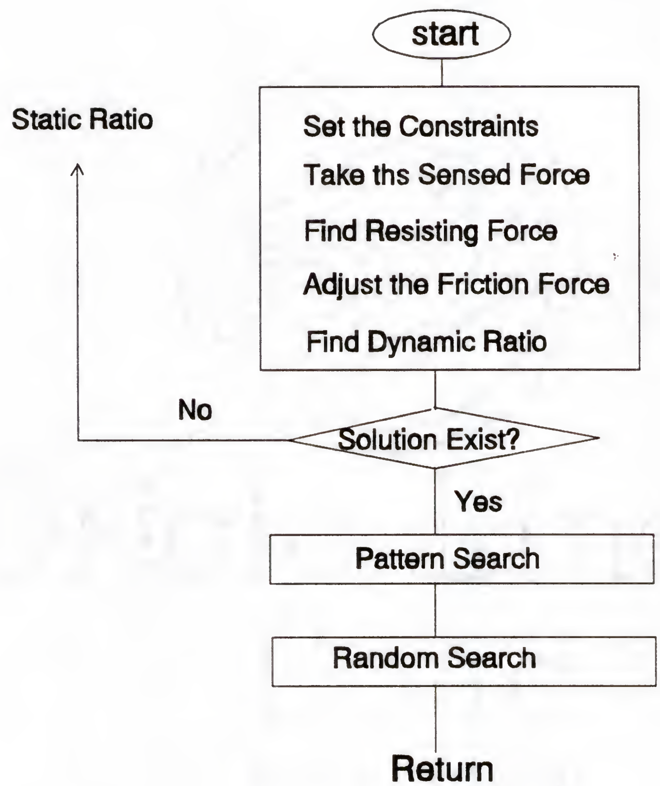

Figure 3.4 Dynamic Ratio

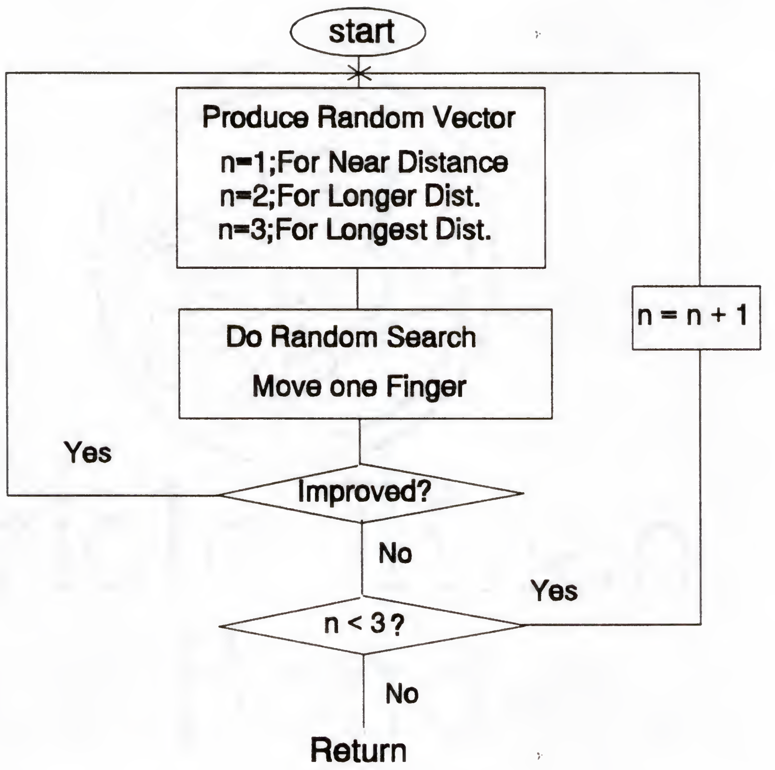

Figure 3.5 Random Search

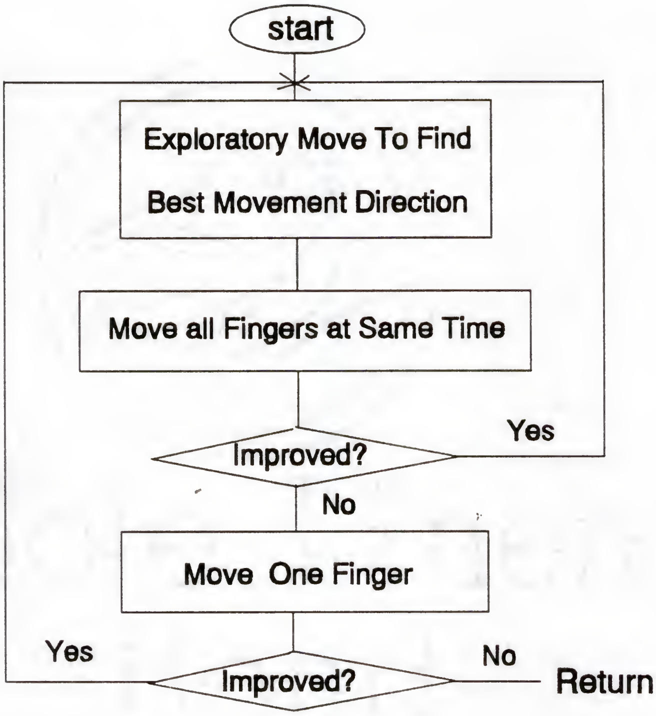

Figure 3.6 Pattern Search

3.6 Sample Results

Sample results from the optimization algorithm based on the previous analysis for a cube and a sphere are given in the following.

The inputs to the program are the geometry of the object, and the external forces. The optimal contact points as well as the magnitude and the orientations of the finger forces are then calculated. Illustrative cases are given below.

3.5.1 Cube

Statement of the Problem

Given : A cube is grasped by three fingers. Two of them can move on y-z planes and the third on an x-z plane as shown in Figure 3.7. The coefficient of friction between the object and fingers is assumed to 0.4. The maximum external forces and moments expected during the task are given as :

$$W_e = (0,-50,-10,-0.5,0,0)^t \tag{3.27}$$

Find : The best three contact points on the planes.

To Minimize : The maximum finger normal force required to move the object without slip.

Optimum grasp for the weightless object taking into consideration the entire dynamic movement. The starting condition and the evaluated optimal contact points and finger force ratios are tabulated as follows.

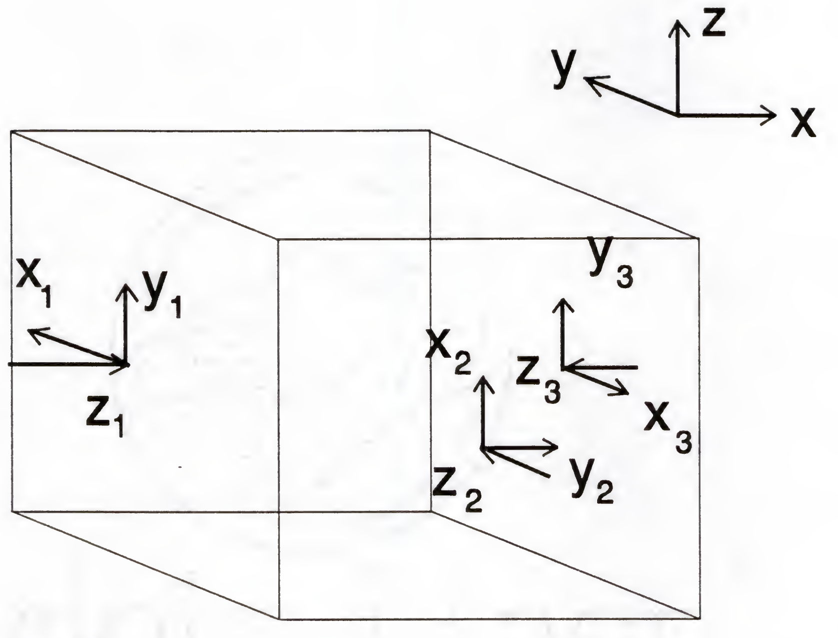

Figure 3.7 Cube Gripped by 3 Fingers

Starting and Optimal contact positions

Table 3.1 Starting Position

$P_1 = [\ -5.0,\ 0.1,\ 0.0\ ]^t$ [cm]

$P_2 = [\ 0.0,\ -5.0,\ 0.1\ ]^t$ [cm]

$P_3 = [\ 5.0,\ 0.2,\ 0.0\ ]^t$ [cm]

Table 3.2 Optimal Contact Position

$P_1 = [\ -5.0,\ 0.1,\ 0.0\ ]^t$ [cm]

$P_2 = [\ 0.0,\ -5.0,\ -0.8975]^t$ [cm]

$P_3 = [\ 5.0,\ 0.1,\ 0.0\ ]^t$ [cm]

Optimum gripping forces to grasp a weightless object with coefficient of friction 0.4 is as follows.

Table 3.3 Optimum Forces (Friction Coefficient = 0.4)

$F_1 = [\ -0.395599,\ -0.059175, 1.00\ ]^t$ [N]

$F_2 = [\ 0.118350,\ 0.0,\ 0.791197\ ]^t$ [N]

$F_3 = [\ 0.395599,\ -0.059175, 1.00\ ]^t$ [N]

<u>Finger Forces corresponding to the Maximum External Forces.</u> The optimal equilibrium forces necessary for resisting the maximum external forces without considering the frictional constraints are found to be :

Table 3.4 Equilibrium Forces

F_1 = [11.111111, 6.255787, 27.777778]^t [N]

F_2 = [-2.511574, 0.000000, 27.777778]^t [N]

F_3 = [-11.111111, 6.255787, 27.777778]^t [N]

It can be seen from the table that the frictional constraint is violated at fingers 1 & 3.

<u>Optimum Forces with Frictional Constraints</u> : Without violating the frictional constraints, the equilibrium finger forces required for resisting the external forces throughout the given task are obtained by using the optimum ratio as follows.

Table 3.5 Optimum Ratio And Forces

Optimum Ratio = 74.267064

F_1 = [-29.379953, -4.394751, 74.267064]^t [N]

F_2 = [8.789503, 0.000000, 58.759906]^t [N]

F_3 = [29.379953, -4.394751, 74.267064]^t [N]

3.5.2 Sphere

Statement of the Problem

Given : A sphere(R = 0.33 m) is grasped by three fingers on the surface as shown in Figure 3.8. Since the sphere can not generate moments in any motion, the external forces and moments expected during a task are given as :

$$W_e = (-50,0,0,-0.5,0,0)^t \tag{3.28}$$

The coefficient of friction is assumed to be 0.1 .

Find : The best three contact points on the surface.

To Minimize : The maximum finger normal force to move the object without slip.

Optimum grasp for the weightless object taking into consideration the entire dynamic movement. The starting condition and the evaluated optimal contact points and finger forces are tabulated as follows.

Starting and Optimal Contact Positions

Table 3.6 Starting Position (R = 0.33 m)

Theta1 = 90.0,	Phi1 = 90.0	[Deg]
Theta2 = 180.0,	Phi2 = 90.0	[Deg]
Theta3 = 270.0,	Phi3 = 90.0	[Deg]

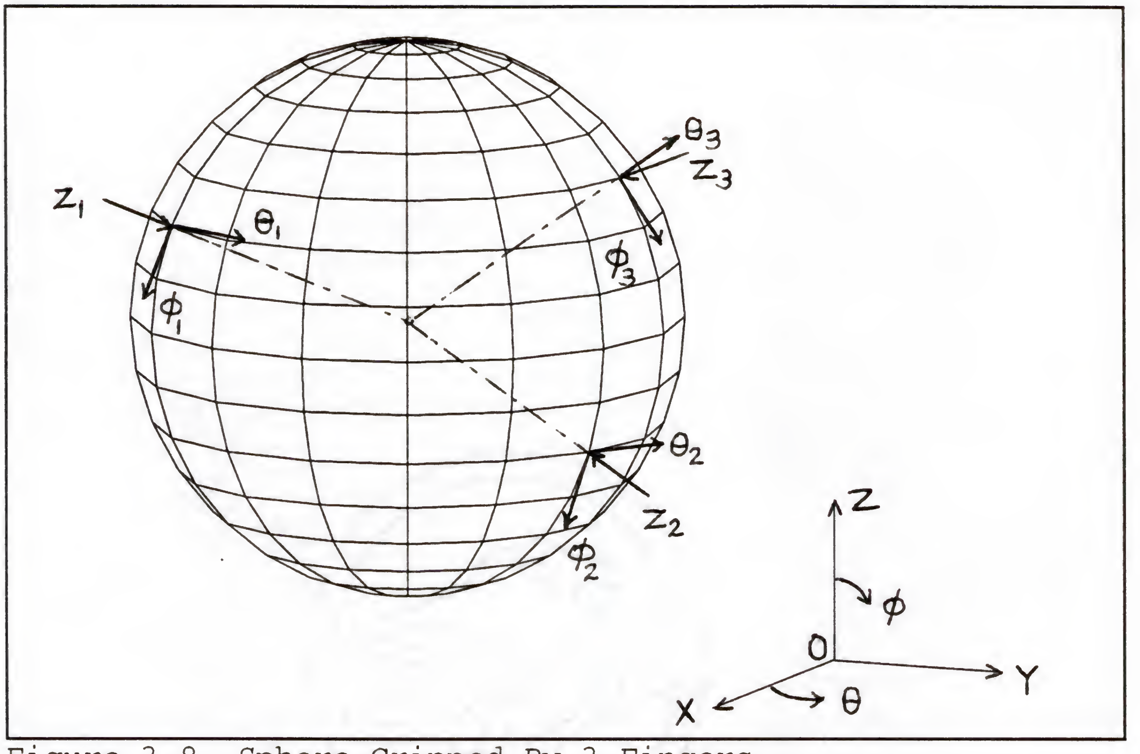

Figure 3.8 Sphere Gripped By 3 Fingers

Table 3.7 Optimal Contact Position (r = 0.33 m)

P_1 =	[Theta1 = -7.3094,	Phi1 = 75.6344]	[Deg]
P_2 =	[Theta2 = 121.0998,	Phi2 = 91.2245]	[Deg]
P_3 =	[Theta3 = 237.6502,	Phi3 = 90.5219]	[Deg]

Optimum gripping forces based on weightless object is as follows.

Table 3.8 Optimum Forces (Friction Coefficient = 0.1)

F_1 =	$[\ 0.063877,\ 0.074035,\ 0.977836]^t$	[N]
F_2 =	$[\ -0.074035,\ 0.066571,\ 1.000]^t$	[N]
F_3 =	$[\ 0.012139,\ 0.074035,\ 0.979772]^t$	[N]

Finger Forces with the Maximum External Forces

The optimum equilibrium forces for resisting the maximum external forces without considering the frictional constraints are found to be as follows.

Table 3.9 Equilibrium Forces

F_1 = [-0.355430, -0.510909, 0.000000]^t	[N]
F_2 = [4.527051, 0.521523, 40.342741]^t	[N]
F_3 = [-4.181873, -1.256450, 40.342741]^t	[N]

It can be seen that the frictional constraint is violated at fingers 1 and 2 in this case.

Optimum Forces with Frictional Constraints : Without violating the frictional constraints, the equilibrium finger forces for resisting the external forces including inertia forces are obtained by using the optimal multiplier as follows.

Table 3.10 Optimum Ratio And Forces

Optimum Ratio = 43.542512	
F_1 = [2.834560, 3.285324, 43.391883]^t	[N]
F_2 = [-3.285324, 2.954099, 44.375434]^t	[N]
F_3 = [0.538666, 3.285324, 43.477785]^t	[N]

CHAPTER 4
IDENTIFICATION OF THE EXTERNAL DISTURBANCES

4.1 Introduction

An important requirement of the computer-based control system is the ability of identifying any external disturbance. To identify the disturbance acting on the object, it is necessary to know the friction forces and moments at all fingertips. If the external disturbance can be identified at any time, it is always possible to calculate the necessary reactions at the grip which accommodate the disturbance. This requires a sensor system which can continuously monitor the magnitude of normal force, and magnitude and direction of the friction force precisely. Also, since it takes some time to calculate the external disturbance from the sensed information, it may be difficult to use these sensor systems in real time control. It is envisioned that a computer based sensor system which is fast enough and to sense the normal and friction force can be developed for use in real time control.

A real time control strategy is highly dependent on the sensor system. If the sensor system is fast enough to monitor the normal forces and friction forces precisely, the computer control system can calculate the exact rotation of the wrist and the normal force which are required to resist the

disturbance. Piezo electric sensors at fingertip coupled with friction force tactile tangential sensors can be utilized in this type of sensor system.

Since it is difficult to directly sense the direction and magnitude of the friction force, moment sensors at the wrist can be used to identify the disturbing force and moment acting on the object at any instant. In this case, tactile sensors at the fingertip can be used to detect the direction of the friction force at each fingertip. Normal force sensor and tactile tangential sensor at fingertip and moment sensor at wrist are utilized in the different sensor systems considered in the strategy discussed in this section.

The piezo electric meter considered as a normal force sensor has good accuracy and fast response. If it is possible to use the normal force information for the control only, the grip can be easily adjusted to accommodate the disturbance in real time control and it would consequently be the best approach to implement in a practical system.

Several computational algorithms are developed for control. The first uses all the information from a comprehensive sensor system which can sense the frictional force as well as the normal forces. The Second is based on the normal force information only to resist the external disturbance.

4.2 Sensing System

4.2.1 A-Full Identification

Initial State. When an articulated hand with n fingers is performing a preplanned operation, each finger exerts a normal force with the ratio which is necessary to sustain the stability of the object when subjected to expected forces. This force ratio is obtained by using the optimization procedure as explained in previous chapter. The fingers with this force ratio do not produce any net force and moment on the object while no unexpected disturbance is occurred.

$$\sum_{i=0}^{n} T_i \cdot f_i = 0 \quad (i = 1,2,\ldots,n) \tag{4.1}$$

T_i is the 6 x 6 transformation matrix which transforms i_{th} fingertip to the center of the object coordinate system.

$$T_i = \begin{bmatrix} R_i & 0 \\ P_i & R_i \end{bmatrix} \tag{4.2}$$

$$R_i = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix}_i , \quad P_i = \begin{bmatrix} l_z y - l_y z & m_z y - m_y z & n_z y - n_y z \\ l_x z - l_z x & m_x z - m_z x & n_x z - n_z x \\ l_y x - l_x y & m_y x - m_x y & n_y x - n_x y \end{bmatrix}_i \tag{4.3}$$

If the position vectors of the fingertips, [x , y and z in equation (4.3)], and the orientations of the fingertip

coordinate systems, [$l_x, l_y, l_z, m_x, m_x, .., n_y, n_z$ in rotation matrices] are known, these transformation matrices can be written as shown in equation (4.2) and (4.3).

$$f_i = [f_{x_i}, f_{y_i}, f_{z_i}, 0 , 0 , m_{z_i}]^t \tag{4.4}$$

The force wrench of the i_{th} finger f_i is defined in equation (4.4) where f_{x_i}, f_{y_i} are the friction force components in the x , y direction of the finger coordinate system and f_{z_i} , m_{z_i} are the normal force and moment components in and about the z direction of the finger system respectively. Since the fingers can not resist x and y components of moments, these values are set equal to 0.

<u>When an Unexpected Disturbance Occurs.</u> If some disturbance has taken place, the force wrenches of the fingers will change as a result of the disturbance as shown in equation (4.5).

$$\sum_{i=0}^{n} T_i \cdot [f_i + \Delta f_i] + f_{ext} = 0 \tag{4.5}$$

where

$$\Delta f_i = [\Delta f_{x_i}, \Delta f_{y_i}, \Delta f_{z_i}, 0 , 0 , \Delta m_{z_i}]^t, \tag{4.6}$$

$$f_{ext} = f_{\exp} + f_{unexp} = \begin{bmatrix} f_{x_o} + f_x \\ f_{y_o} + f_y \\ f_{z_o} + f_z \\ m_{x_o} + m_x \\ m_{y_o} + m_y \\ m_{z_o} + m_z \end{bmatrix} \tag{4.7}$$

The normal force change Δf_{zi} will be sensed from the normal force sensor at the i_{th} fingertip. The change of frictional forces at each fingertips in equation (4.6) will be also sensed if there are tactile tangential sensors. The equilibrium equation (4.5) can be rewritten as

$$\sum_{i=0}^{n} T_i \cdot \delta f_i = - f_{ext} \quad (i = 1,2,\ldots.,n). \tag{4.8}$$

where

$$\begin{aligned} \delta f_i &= [\delta f_{x_i}, \delta f_{y_i}, \delta f_{z_i}, \ 0 \ , \ 0 \ , \delta m_{z_i}]^t \\ &= [f_{x_i} + \Delta f_{x_i}, f_{y_i} + \Delta f_{y_i}, \ldots\ldots\ldots, m_{z_i} + \Delta m_{z_i}]^t \end{aligned} \tag{4.9}$$

To be on the conservative side, it is assumed that the friction moment δm_{z_i} has a maximum value proportional to the normal force of the finger. If the finger has constant contact area at the tip, the maximum moment it can resist will increase proportionally when the normal force at the contact is increased.

Since all the components of the finger forces and moments are known, the external disturbance f_{ext} can be readily obtained.

If the tactile sensors at each fingertip sense the direction of the friction force only, the moment sensor at wrist is needed for the exact identification of the external interruption. In this case, equation (4.9) can be rewritten as equation (4.10).

$$\delta f_i = [\delta f_{r_i} \cos(\alpha_i), \delta f_{r_i} \sin(\alpha_i), \delta f_{z_i}, 0, 0, \delta m_{z_i}]^t \quad (4.10)$$

The magnitudes of friction forces f_{ri} and external force f_x, f_y and f_z are the unknowns and sensed wrist moments m_x, m_y, m_z, the directions of the friction forces at each fingertips α_i and the normal forces f_{zi} are the knowns in this case. If the number of the fingers is 3, there are 6 unknowns in 6 linear equations as before. The exact disturbance will be readily evaluated.

4.2.2 Approximate Identification

Since it is generally difficult to incorporate a tactile sensor which can sense the exact magnitude and direction of the friction force at the fingertip, an approximate identification procedure is attempted in this section. If the

sensor system consists of normal force sensors at the fingertips and moment sensors at the wrist without having any tactile sensor for the friction, and the manipulator has three fingers, the number of unknowns are 9 in 6 linear equations in this case. They are three f_{xi} , three f_{y_i} and f_x , f_y and f_z . There might be many solutions of the unknowns because they are underdetermined. If any feasible solution is obtained, it can be applied to the control logic which can adapt the solution to be under the control. Since the purpose of this is to get approximate external disturbance, possible maximum can be obtained after extensive computation.

Another viable method is introduced again. Since it is under determined, several more usable linear equations can be acquired it will have a unique solution. It is able to be assumed that the friction force isn't change a lot with very small rotation of the object while under the same interruption. By rotating very small angle about x,y and z axis, 3 sets of the linear equations can be obtained. To get an approximate value of the disturbance, it is possible to choose one completed set of the linear equations from the 3 sets when same frictional terms are used. Since many possibilities should be considered in this procedure it takes time. In a real time control, all the calculations at each time step must be completed in a very short time because all the proper reactions also should be accomplished in the time step. If a fast enough computer and reactor such as a quick

servomotor are developed, these approximate methods can be considered to identify the external interruption approximately.

4.2.3 Analysis of The Change in Normal Forces and The Ability of The Grasp

Analysis of The Change of The Normal Forces. The following information can be determined from the analysis of the sensed normal forces. In the initial state, each finger has a normal force and friction force which do not produce any net resultant force and moment on the object if there is no unexpected disturbance. When some external disturbance occurs, the normal force sensors detect the change in the normal forces due to the disturbance which manifests itself as the changed reactions normal to the contact areas. This change quantifies the force ratio of the fingers which is necessary to resist the external disturbance without slip if this is possible. The change due to the unexpected disturbance can be obtained since the expected disturbance is known.

Other useful information is that the sum of the normal forces required to resist the disturbance and the sum of the moment due to these normal forces for all the fingers about the center of the object(Origin of the coordinate system). From this information, a rough estimation of the external

$$\begin{bmatrix} f_{x_o} \\ f_{y_o} \\ f_{z_o} \\ m_{x_o} \\ m_{y_o} \\ m_{z_o} \end{bmatrix} = \sum_{i=0}^{n} \begin{bmatrix} R_i \ , & 0 \\ 0 & , P_i \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ f_{z_i} \\ 0 \\ 0 \\ f_{z_i} \end{bmatrix} \tag{4.11}$$

change is possible. This will be shown later. P_i in equation(4.11) is the same 3 X 3 matrix in equation (4.3).

Evaluation of ability of the grasp to resist disturbance. The optimal grip evaluated before the disturbance has special characteristics, such as which direction of the external disturbance is most or least resistible. This information which is known before starting the correction indicates which rotating direction is the most effective and the one which is most detrimental.

Table 4.1 : Characteristic of the Optimal Configuration

AXIS	FORCE (20.0 N)	MOMENT (0.5 N-m)
+ X	125.188	53.598
+ X	can't resist	1367.440
+ Y	33.484	104.846
- Y	24.198	104.846
+ Z	83.890	can't resist
- Z	45.071	79.884

Table 4.1 shows one example of this for the case of a cube held with three fingers. The table shows results obtained after the optimal of the grasp by applying arbitrary forces and moments to evaluate the ability of this grasp to resist unexpected disturbances. From the table, the grasp in this case cannot be maintained in the case of some disturbing force in negative X axis direction or a moment about positive Z axis.

4.3 The Utilization of The Sensed Information

4.3.1 With A Comprehensive Sensor system

Extrapolation of the sensed disturbance can be considered for the next step if the sensor system can sense the friction forces as well as the normal forces completely at fingertips. Polynomial approximation is one of the most effective techniques without expensive calculation time. If some unexpected disturbance is sensed for the first time, a two point linear approximation can be used and a three point quadratic approximation, four point cubic approximation can be used for the next sequences.

When the sensed maximum normal force is greater than the current normal force in a finger, increasing the force ratio of the grip immediately to the new level corresponding to the increased force in that finger can prevent the object from slipping. Furthermore, it is possible to improve the grasp

for the next time by changing the orientation of the object with small angle for the extrapolated value . If the checked normal force in advance decreases with a small rotation, the grasp is judged to be improving. If it is increasing the grasp will be worsening. Following this procedure, it is possible to avoid wrong rotation of the object for the next step. It should be noted that the resisting normal force will increase sharply if the object rotates in the wrong direction.

4.3.2 With A System With Normal force sensors only

The change of the required normal forces and the change of the corresponding moment of these forces about the center of the object can be treated as a trial additional external disturbance for the next time. Since the sensed normal force implicates some part of the external disturbance and the obtained optimal configuration of the grasp was obtained based on minimized maximum friction force, this can be a possible way to avoid wrong rotation of the object for the next step.

The inherent characteristic of the initial grasp is an indicator of which rotating direction and axis of rotation will improve the grasp and which direction and axis will weaken the grasp based on the trial disturbances applied during the evaluation process before the movement starts(as illustrated in Table 4.1).

<u>Estimating the Disturbance.</u> The solution vector can be estimated, by using some basic condition. One important step is to examine if any of the fingers is slipping or shall slip soon at this moment. One way is to use emergency factor defined in equation (4.12) below. The terms in the numerator of the equation represent the frictional force and moment acting on the object currently. The denominator represents the normal force necessary to prevent the object from slipping at this instant.

$$U_i = \frac{\sqrt{f_{x_i}^{\ 2} + f_{y_i}^{\ 2}} + A\,|m_{z_i}|}{f_{z_i} \cdot \mu} \tag{4.12}$$

If U_i is greater than 1 the finger is in a state of slip. If it is much smaller than 1, such as less than 0.5, the finger will not slip at this instant. If the finger is slipping, immediate reaction, such as increasing the normal force, will be required immediately.

$$Mar = \frac{f_{z_{limit}} - f_{z_{max}}}{f_{z_{limit}}} \tag{4.13}$$

There can be an upper normal force limit for avoiding damage to the finger or the object. Another factor ***Mar*** defined in equation (4.13) can be used for determining how much of the finger normal force can be added without violating

the upper limit. Where $f_{z_{max}}$ in equation (4.13) is the maximum normal force of the fingers. If ***Mar*** is close to 0, the normal force should not be increased.

CHAPTER 5
CONTROL ALGORITHM

5.1 Introduction

In the previous chapter, several computational algorithms to identify the external disturbing force and moment with different degrees of certainty were introduced. This chapter addresses the question of how to control the manipulator to perform the desired motions in the presence of the unexpected disturbances. The control algorithms presented in this chapter are based on the previously discussed different identification algorithms. The control for the case of a comprehensive sensor system is discussed first, another useful control which relies only on the input from normal force sensors is discussed next.

An important factor which is taken into consideration in developing these algorithms is that all computations and reaction evaluations should be completed within the period needed for data acquisition by the sensor system. Despite the fact that highly developed computer systems can provide a great advantage in computation time, there appears to be no suitable control algorithms available in the published literature because of the long computation time which would be required for the task under consideration. The generalized

optimization algorithm for minimizing the maximum finger normal forces when grasping a moving object developed in chapter 3 can provide a considerable advantage in computation time. By using the force ratio the required normal forces of the fingers which are necessary to maintain the stability of the object can be estimated immediately when an external disturbance occurs.

The possible corrective actions which are considered are increasing the finger normal forces and changing the orientation of the object by arm or wrist rotations. Since the starting optimal force ratio was obtained based on the expected external changes through out the entire motion, it generally will generate more finger forces than is necessary to resist the expected disturbance at any time. Accordingly, when a relatively small unexpected disturbance occurs the fingers may be able to sustain the object safely. It can also be assumed that when some disturbance occurs for the first time, the object will not start to slip immediately, and the fingers can keep the stability of the object by adding the required normal forces immediately if the sensor system can monitor very small changes as the disturbance progresses. By the time it reaches the maximum, it is possible to anticipate the progression. After adding the finger normal forces instantly to resist the external force and moment, the grasp can be improved for resisting stronger disturbances from the same direction by simply changing the orientation of the

object about appropriate axes.

If the expected disturbance vanishes, the grasp can then be returned to the original condition and be prepared for the next emergency.

In the case of relying on normal force sensors only, since the change of the normal force may be radical, the extrapolation can not guarantee proper values for the next step. The change of the required normal forces and the change of the corresponding moment of these forces about the center of the object can be treated as a trial additional external disturbance for the next time as explained in previous chapter. If the unexpected disturbance is stronger, the sensed normal force differential will constitute a larger segment of the total signal and this method becomes more reliable under such condition.

5.2 Control with a Comprehensive Sensor System

A comprehensive sensor system detects all the information about the change in normal and the tangential friction forces at the fingertips resulting from any external disturbance. Such a system would incorporate tactile sensors which can sense the magnitude and direction of the friction forces and normal force sensors at fingertips. If the tactile sensors are not sufficient for evaluating the direction and the magnitude of the frictional force at the fingertips, moment sensors at the wrist would be needed to supplement the

shortage of the information. Since the optimal grasp is a preplanned operation, the expected external force and moment can be predicted completely and the grip can be prepared to accommodate all the known external change when there is no unexpected external disturbing force or moment. Since a unexpected disturbance can be calculated from the detected information as in equation (4.6), it also is possible to foresee the best orientation which improves the grasp by virtual change of the orientation about all the axes to positive and negative direction with the extrapolated estimates of the disturbance. If some unexpected disturbance is detected, prompt adding of the required normal forces of the fingers is necessary first and finding the best orientation of the object is the next action. If the unexpected external disturbance is changing it is necessary to forecast the disturbance which will occur next by extrapolating the detected change.

The main advantage with this complete sensor system is that the grip can resist all the resistible disturbance by increasing the exact amount of the required normal forces and changing the orientation of the object in a deterministic manner without the possibility of estimation errors which can be the case if the manipulator is not equipped with a comprehensive sensor system.

5.3 Control with Normal Force Sensors Only

A sensor system which has normal force sensors only can detect only information about the normal forces at the fingertips which result from the external disturbance. Since this is a preplanned operation, the expected external force and moment can be resisted completely. The grasp, however, is not expected to be able to accommodate any temporary external change without a comprehensive sensor system if there is some unexpected external disturbance. Since the unexpected external disturbance can not be calculated deterministically from the detected normal force information, it is not there fore possible to foresee the best orientation of the object to improve the grasp and prevent slip.

The sum of the normal forces required to resist the disturbance and the sum of the moments due to these normal forces for all the fingers about the center of the object (Origin of the coordinate system) as obtained from equation (4.13) is considered. From this information, a rough estimation of the external change is possible. A negative change of these values can be treated as a known disturbance so that it is possible that the sum of the these values and the expected disturbance can be used as the completely detected disturbance as in the case of the control with a comprehensive sensor system for the next step.

The change of the normal force due to the unexpected disturbance can be instantaneously detected. If some change

is detected, it is necessary first to immediately apply the required normal forces of the fingers(as long as it is below the allowable limit), and then finding the best orientation of the object as the next step.

In this case it is possible to foresee a better (not the best) rotation of the object to improve the grasp by incremental rotating the wrist about all the axes in the positive and negative directions. Consequently it is possible to avoid unacceptable large increase of the normal forces.

5.4 Possible States

After the preplanned operation is started, it is assumed that an unexpected disturbance can occur at any time. It can occur before or after the expected forces take place. According to the time when an unexpected disturbance occurs, the control of the grasp should be prepared for the change. When the preplanned operation starts it can be assumed that expected and unexpected disturbances has not yet occurred. At this state, the grasp prepares for the expected force and moment which will occur during the operation by generating the required normal forces according to the precalculated force ratio from the generalized optimization procedure. The sensor system starts monitoring every change of the sensed information. If the end effector has a comprehensive sensor system the information will be used in the extrapolation for

the next step. If the fingers have normal force sensors only, the change of the normal forces will be used for the next step. Algorithms from this initial state to other states should be activated at this moment. All the events which can take place during the operation and the control algorithms according to these events are as follows (Figure 5.1).

Initial State : After the start of the operation, no disturbance has occurred yet.

State A : From the Initial state, some unexpected disturbance is detected while no expected disturbance occurred.

State B : From the Initial state, some expected disturbance is detected yet no unexpected disturbance occurred.

State C : From the state A or B, some expected and unexpected disturbance are detected.

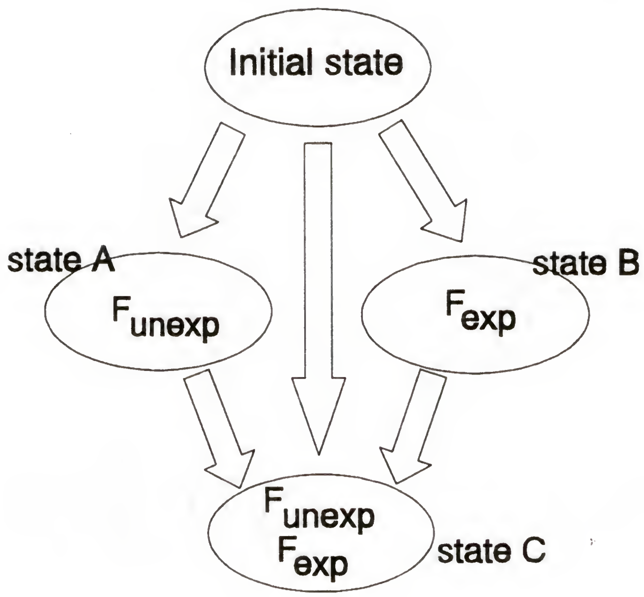

Figure 5.1 Possible States

5.5 Control Algorithms

The control algorithms for these states are as follows.

5.5.1 Algorithms for the Initial State

. Grasp the object with the initial ratio and start monitoring the sensed information.

. If unexpected disturbance is not sensed, keep the original orientation even though the expected disturbance has occurred.

. When any unexpected disturbance is detected.

1. Change the force ratio instantly to generate the appropriate normal forces.
2. Anticipate the next disturbance from the monitored information, calculate the normal force ratio and find the rotation axes and directions which will improve the grip for the next step.
3. Rotate the object about the axes for the next step.

. Add the information about the expected disturbance just before the disturbance occurs for the next step.

The first algorithm can be used at any time if some exist disturbance has disappeared. This is especially important when the grasp is in the state A and state C. Without this algorithm the manipulator can not be prepared for the next

expected disturbance from these states. Once the sensor system starts monitoring the sensed information, it will last until the manipulator finishes the operation.

The second algorithm is for continuing the state or for reaching state B from the initial state. Since the grasp has already been prepared well for the expected disturbance, the manipulator can sustain the stability of the object without changing the force ratio or the orientation of the object.

The third algorithm is for reaching state A from the initial state. This can also be used when the disturbance is changing. It is the most difficult situation when the state changes from the initial state to the state A. If the grasp can not resist this unexpected disturbance when it is detected for the first time, the manipulator can not perform the operation successfully. This can occur when the disturbance is too large or it is from a direction the grasp cannot resist. The human hand also cannot hold an object in this type of situation.

The fourth algorithm is for reaching state B from this initial state. Because of the conditions for the initial state, the grasp can be prepared for a sharp increase of the required normal forces due to the expected disturbance during this initial state.

5.5.2 Algorithms for the State A

. If the detected unexpected disturbance is variable.

1. Change the force ratio instantly to generate the appropriate normal forces.

2. Anticipate the next disturbance from the monitored information, and calculate the normal force ratio and find the rotation axes and directions which will improve the grip for the next step.

3. Rotate the object about the axes for the next step.

. If the detected disturbance is not changed from the previous step, keep the current normal force ratio and orientation of the object.

. Add the information of the expected disturbance just before the disturbance occurs for the next step.

. If the detected disturbance ends return to the initial state.

The first algorithm is exactly the same that the third one in the initial state. If the unexpected disturbance is changing rapidly and the grasp can not accommodate it, the system may become unstable.

The second algorithm is always effective when the sensed unexpected disturbance does not change.

The third algorithm is for reaching state C from state A and this algorithm is always effective if the expected disturbance is beginning to occur.

The fourth algorithm is for reaching the initial state from state A.

5.5.3 Algorithms for the State B

. If no unexpected disturbance is sensed, keep the original orientation even though expected disturbance has occurred.

. When any unexpected disturbance is detected.

1. Change the force ratio instantly to generate the appropriate normal forces.

2. Anticipate the next disturbance from the monitored information, calculate the normal force ratio and find the rotation axes and directions which will improve the grip for the next step.

3. Rotate the object about the axes for the next step.

Algorithms in this state are the same as those for the initial state because the operation is preplanned and the grasp has been prepared for this state. And the second algorithm is for reaching state C from state B.

5.5.4 Algorithms for the State C

. If the unexpected disturbance is changing.

1. Change the force ratio instantly to generate the appropriate normal forces.

2. Anticipate the next disturbance from the monitored information, and calculate the normal force ratio and find the rotation axes and directions which will improve the grip for the next step.

3. Rotate the object about the axes for the next step.

. If the unexpected disturbance ends, return to the initial state and perform the algorithms for that state.

. Add the information about the expected disturbance just before the disturbance occur for the next step.

The second algorithm is for the state B and the third algorithm is for reaching state B or the initial state from state C respectively.

All the algorithms corresponding to each state can be combined for the entire control operation which can maintain the stability during the operation for all the circumstances if the disturbances are resistible. Sometimes the manipulator can accommodate some disturbance which are not resistible.

If the system has a comprehensive sensor system and the disturbances are resistible, it is always possible to maintain the stability during the operation.

Several other algorithms can be considered such as :

. Choose the one best rotation for improving the grasp to reduce the reaction time.

. If the system has a comprehensive sensor system and the only current disturbance is unexpected, compare the required normal force with the current orientation and with the original orientation for the disturbance, and then choose the better orientation based on the comparison.

5.5.5 Algorithm for Finding the Best Orientation

The algorithm for finding the best orientation which is required in all the states uses most of the computation time during the procedures. Rotation matrix which represents the orientation of the object at the moment is calculated every step. Since the different order of rotations produces different orientation, it is required to choose the order of rotations. Following is the algorithm for finding the best orientation by trial rotations.

1. Calculate the required normal force ratio for the anticipated disturbance which will occur next time with current orientation.

2. Rotate to the positive and negative direction by a step degree about each axis and check if the required force ratio is increasing.

3. For each axis, choose the value 1.0 if the required force ratio is decreasing with positive rotation, or choose -1.0 if the required force ratio is decreasing with negative rotation, or choose 0.0 if there is no improvement.

4. Make a rotation matrix with a chosen order.

5. Multiply the rotation matrix before the current rotation matrix, and check if it is improving.

6. If it is improving, repeat previous step 5 until it isn't improving.

The control algorithms introduced here are very simple and easy to implement in a real time control. Some results of several practical examples will be shown and discussed in the next chapter.

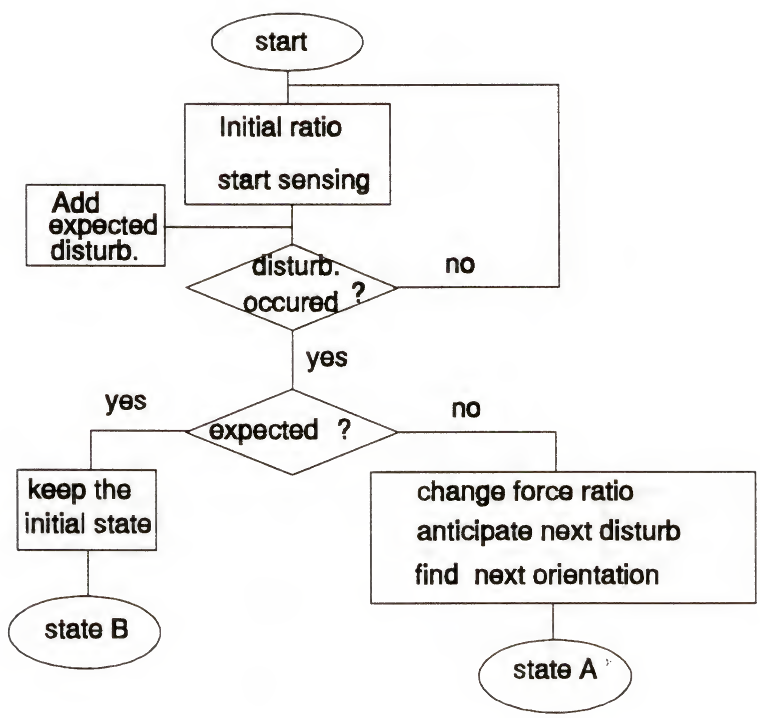

Figure 5.2 Initial State

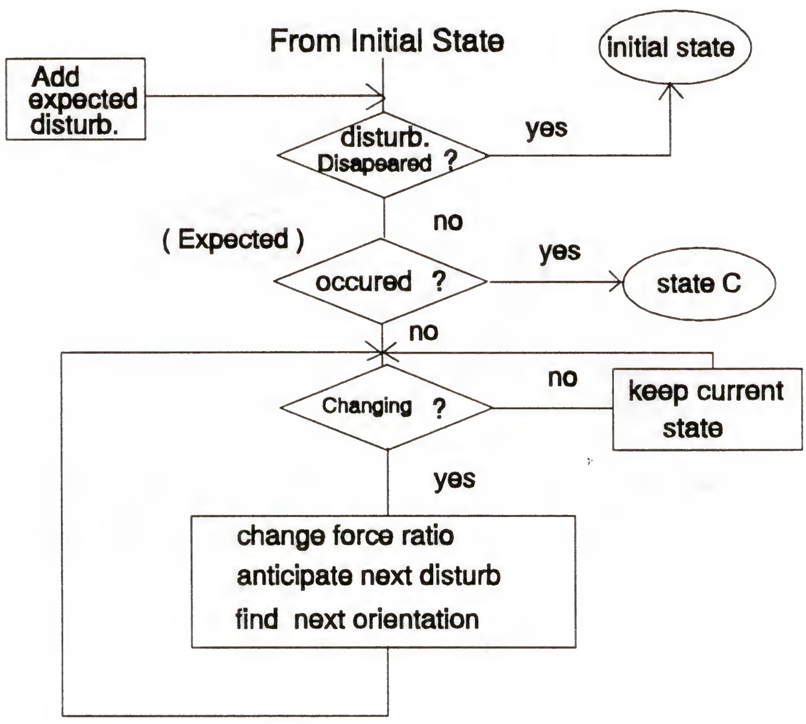

Figure 5.3 State A

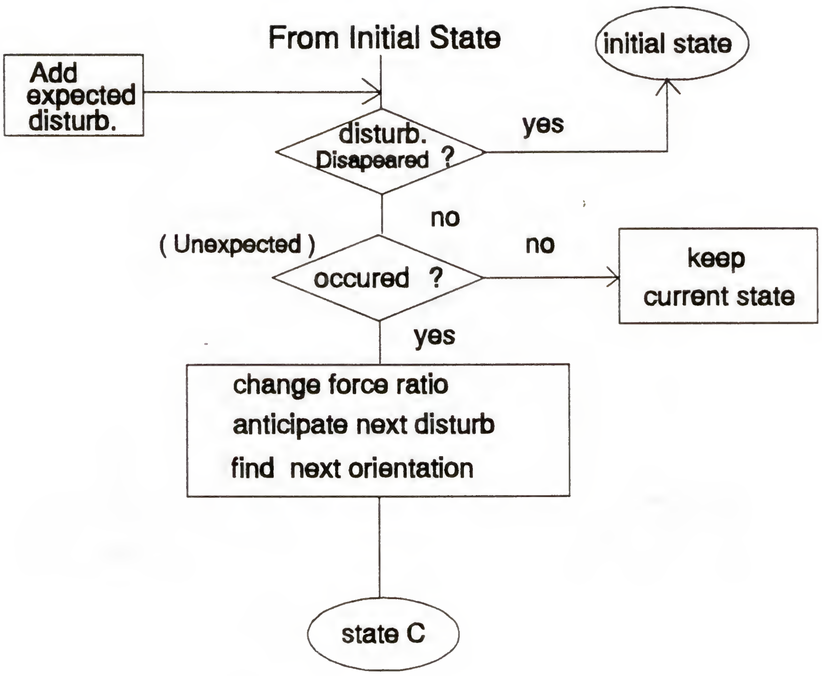

Figure 5.4 State B

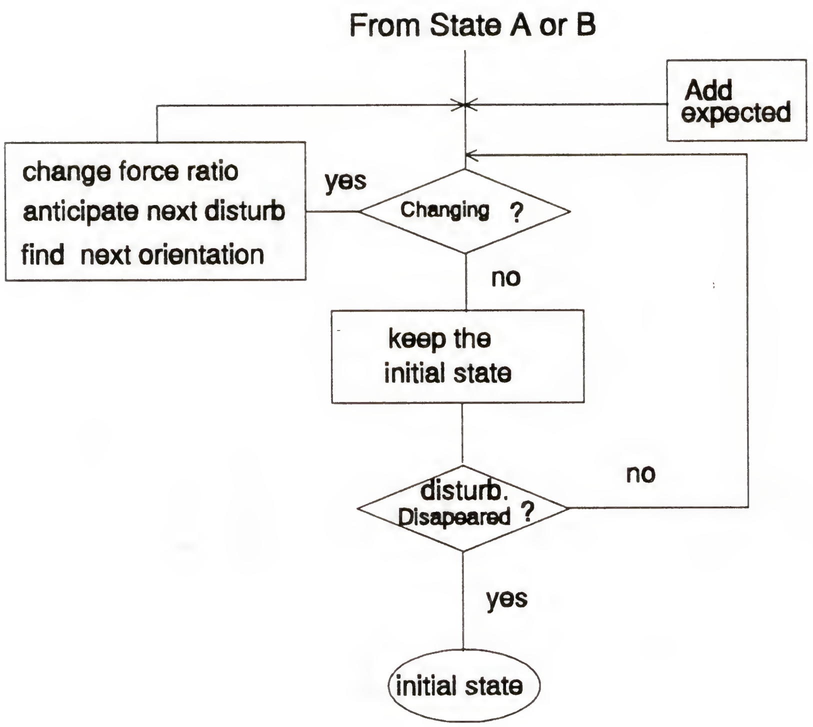

Figure 5.5 State C

CHAPTER 6
SIMULATION RESULTS AND DISCUSSION

6.1 Introduction

To examine the feasibility of these control algorithms, several computer simulation examples are run for illustration and the results are given in Figure 6.9 through Figure 6.58. The assumed situations include all the possible states are explained in previous chapter. In the considered examples, several unexpected pulse disturbances are activated while the manipulator is performing the preplanned operation. The unexpected disturbances are shown in Figure 6.6 to 6.8 and they represent both force and moment disturbances. The data acquisition time increment during the simulation is assumed to be 0.01 sec.

The cube model discussed in chapter 3 is used as the object since the cube is the most difficult one among those models. The optimal positions of the fingers and the required normal force ratio for the three expected disturbances during the preplanned operation are as listed in Table 6.1.

The results for the simulations with comprehensive sensor system will be explained first and those with normal force sensors only will be discussed later.

Table 6.1 Starting Condition

OBJECT : CUBE (0.10 x 0.10 x 0.10) [m]

FRICTIONAL COEFFICIENT : 0.3

THREE EXPECTED DISTURBANCES : FORCE;[N], MOMENT;[N-m]

F_1 = [0.0, -30.0, 0.0, 0.0, 0.3, 0.0 $]^t$

F_2 = [0.0, 0.0, -10.0, 0.0, 0.0, 0.0 $]^t$

F_3 = [-20.0, 20.0, 0.0, 0.5, 0.0, 0.0 $]^t$

OPTIMAL FINGER POSITION FOR THE DISTURBANCES :

P_1 = [-0.05, 0.0, 0.00 $]^t$ [m]

P_2 = [0.0039, -0.050, 0.014 $]^t$ [m]

P_3 = [0.05, 0.0, 0.00 $]^t$ [m]

REQUIRED NORMAL FORCE RATIO = 37.582 N

The assumed unexpected situations for the computer simulations are as follows.

. Situation 1 : Three different unexpected disturbances occur with no expected disturbance. (Figure 6.1)

. Situation 2 : Three unexpected disturbances occur in conjunction with three different expected disturbances respectively. (Figure 6.2)

. Situation 3 : Three unexpected disturbances occur in conjunction with an expected disturbance. (Figure 6.3)

. Situation 4 - 5 : Same as situation 2 and 3 but with higher unexpected disturbances. (Figure 6.4 to 6.5)

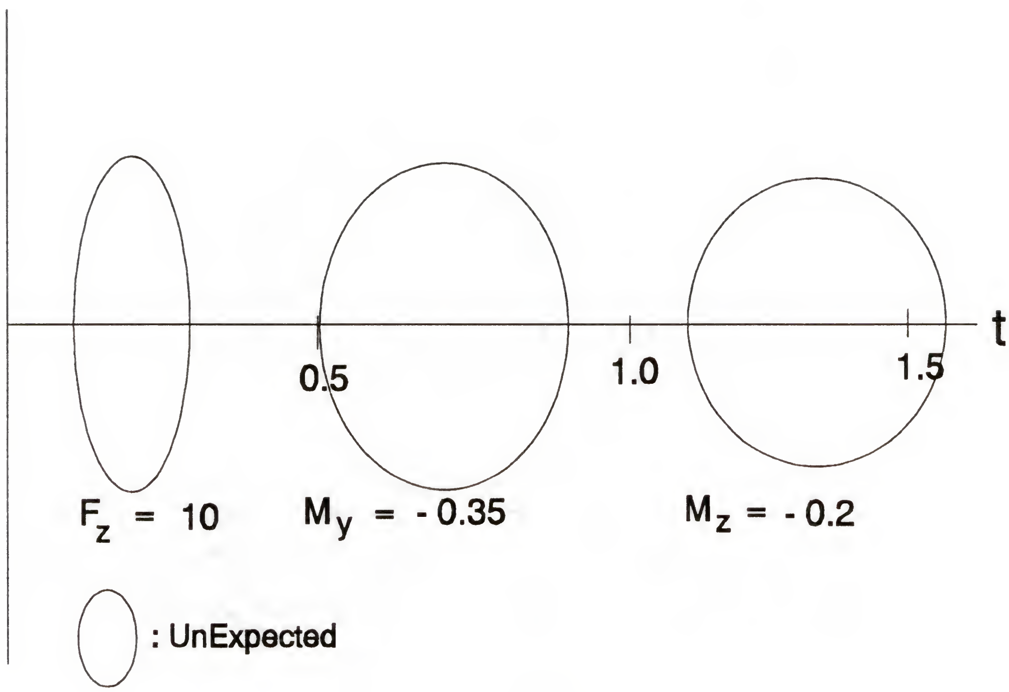

Figure 6.1 Situation 1

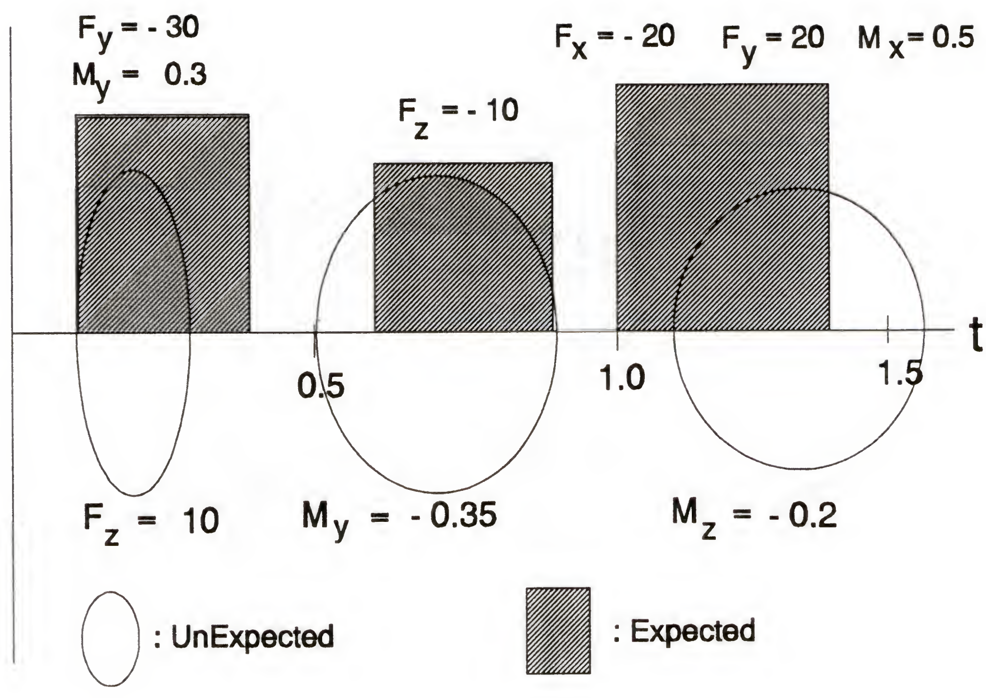

Figure 6.2 Situation 2

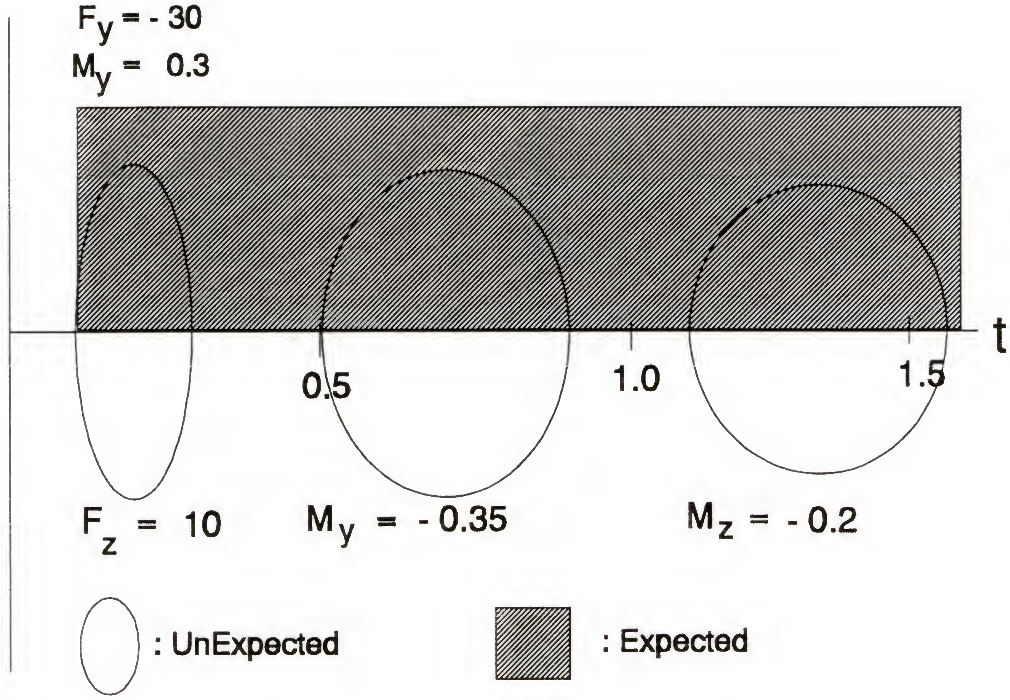

Figure 6.3 Situation 3

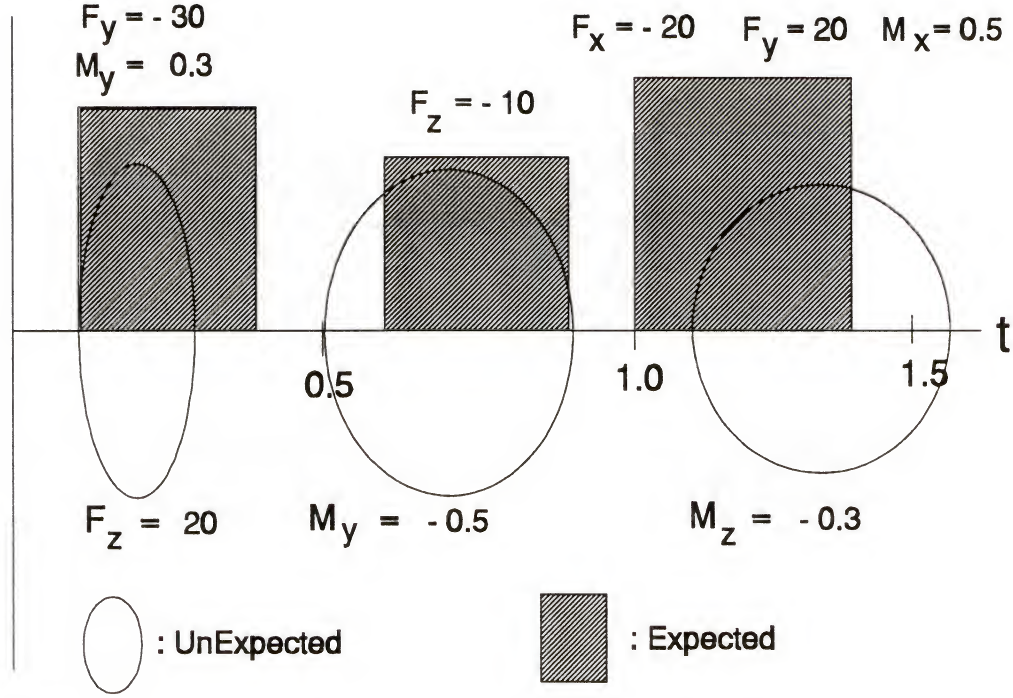

Figure 6.4 Situation 4

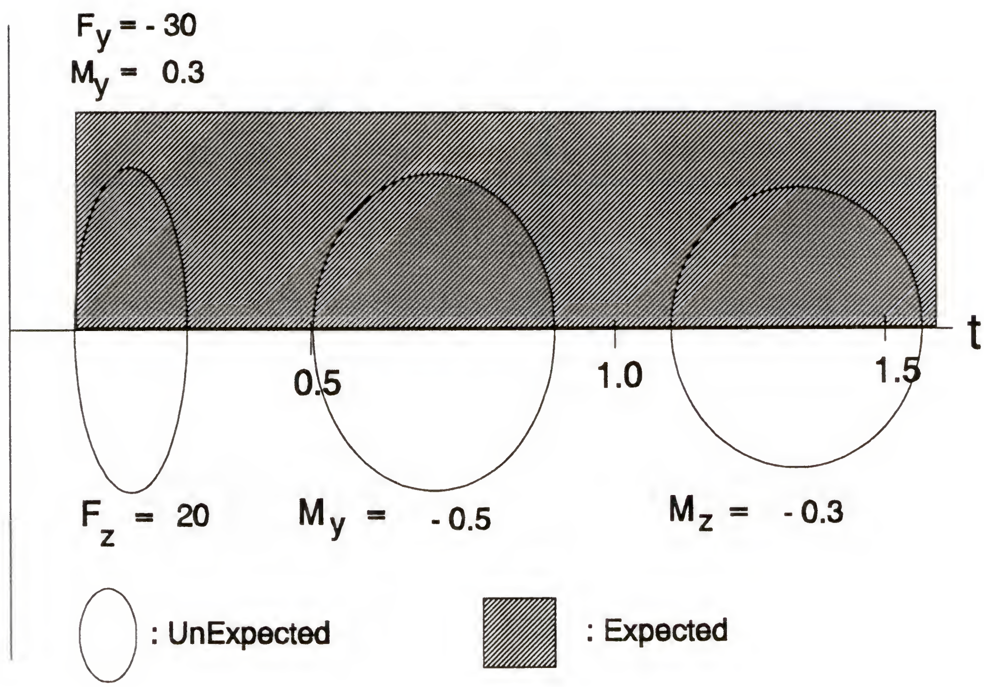

Figure 6.5 Situation 5

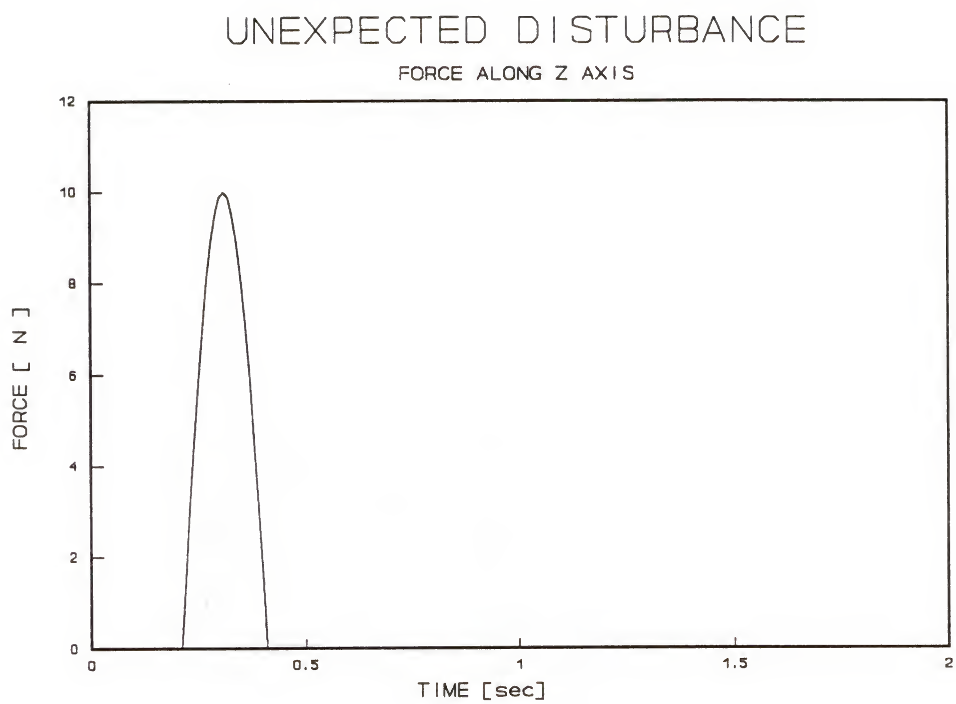

Figure 6.6 The First Unexpected Disturbance

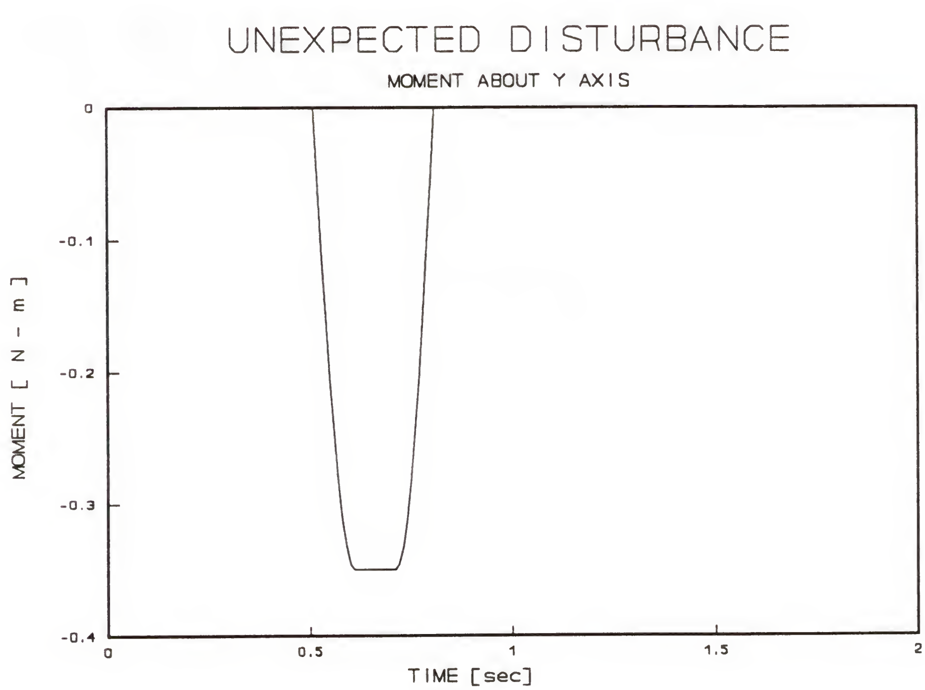

Figure 6.7 The Second Unexpected Disturbance

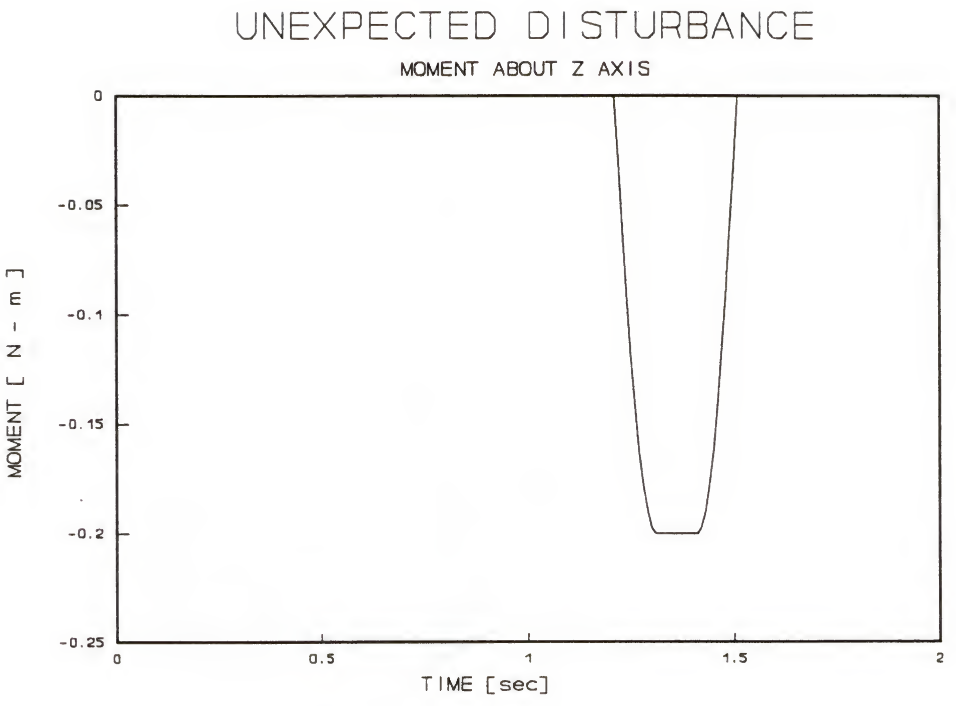

Figure 6.8 The Third Unexpected Disturbance

6.2 Response With A Comprehensive Sensor system

Situation 1 (See Figure 6.9 to 6.14)

Since there is no expected disturbance during the operation, these example show how the system works if an unexpected disturbance occurs without any anticipated forces acting on the object. The system maintains its equilibrium during the force disturbance. The grip holds the object with a very small change of the orientation. The algorithm becomes very active when the negative moment disturbance along y axis occurs. The required normal force is considerably reduced as a result of the moment disturbances when compared with the required normal force in the case without changing orientations as shown in Figure 6.9.

Situation 2 (See Figure 6.15 to 6.20)

Since this situation is assumed to examine the response during the most difficult circumstances, very active reaction is to be expected. The required normal forces are considerably reduced by changing the orientation especially when the second pulse was stimulated. The change of the required normal forces is very high because the disturbing input is changing radically in a very short period.

Situation 3 (See Figure 6.21 to 6.26)

This situation is one of the most common circumstances when considering a real preplanned operation. The grasp is very stable during the entire period of operation. Also the system can sustain the stability of the object without changing the orientation in this case.

Situation 4 (See Figure 6.27 to 6.32)

Since the unexpected disturbances are stronger than situation 2 and 3, the system can not resist any unexpected disturbance without rotating corrections in this case. Because the change of the orientation at each step is rapid, the control should have very quick response characteristics.

Situation 5 (See Figure 6.33 to 6.38)

Even though the unexpected disturbance is larger than the disturbance in the situation 3, the difficulties in the change of required normal forces and of rotating the object are not significantly changed. The difficulties of both cases are almost same.

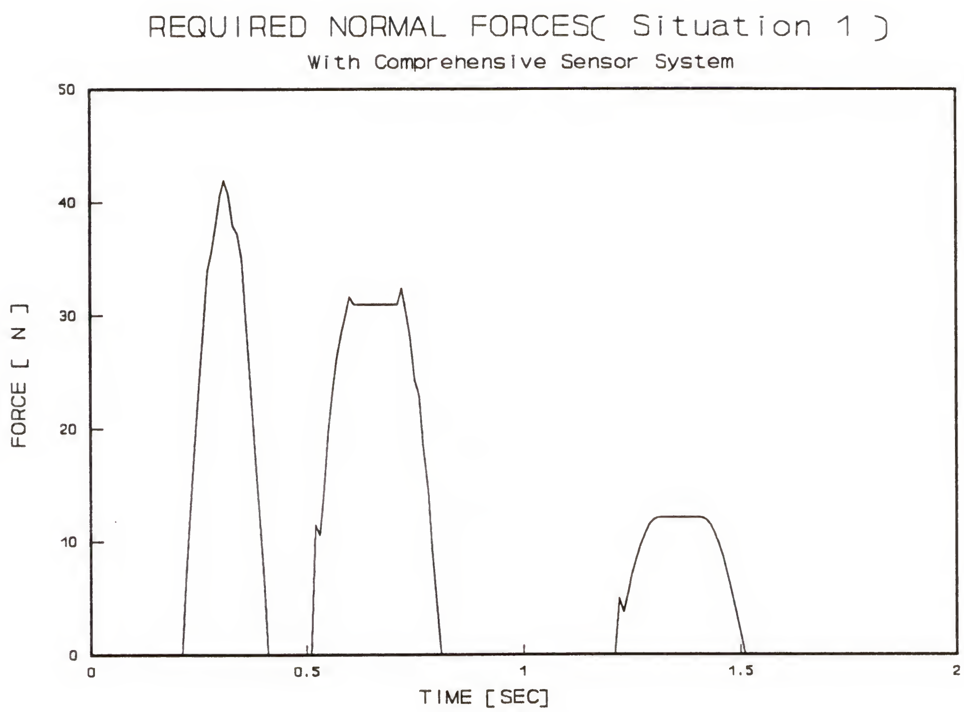

Figure 6.9 Required Normal Forces

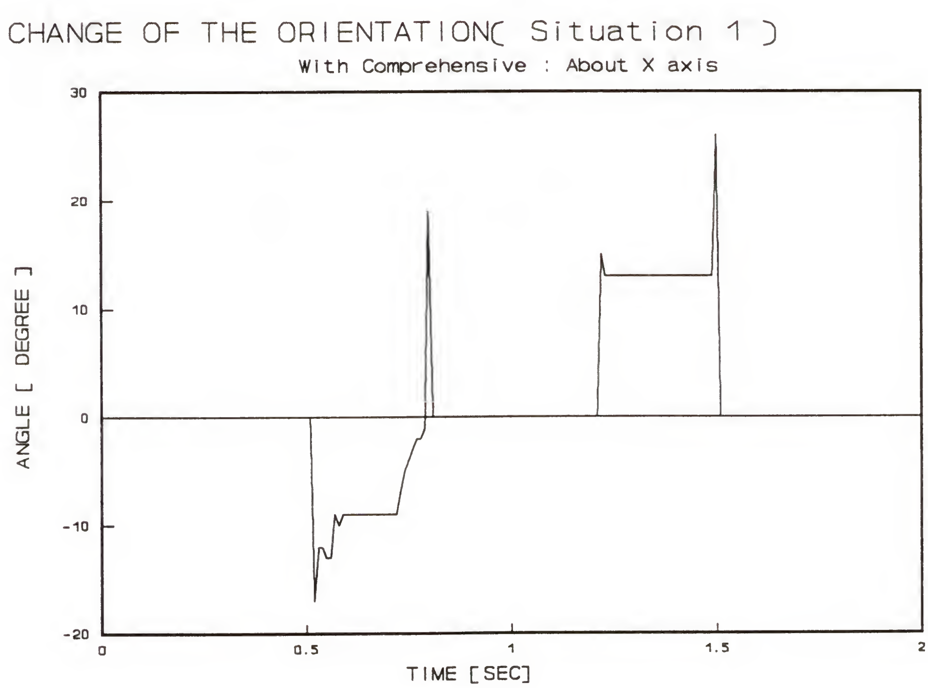

Figure 6.10 Change of The Orientation : About X axis

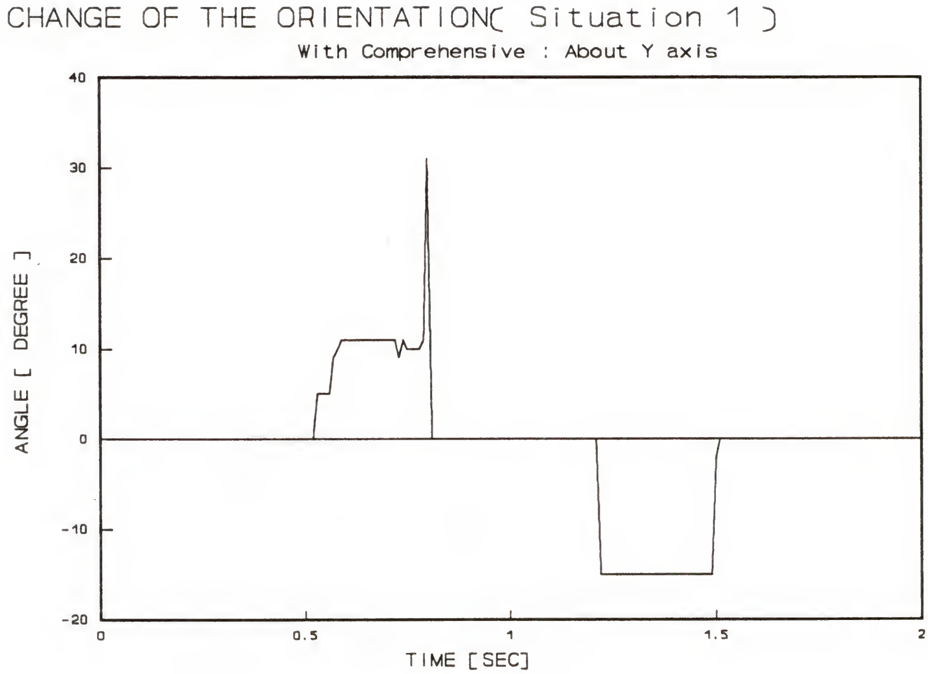

Figure 6.11 Change of The Orientation : About Y axis

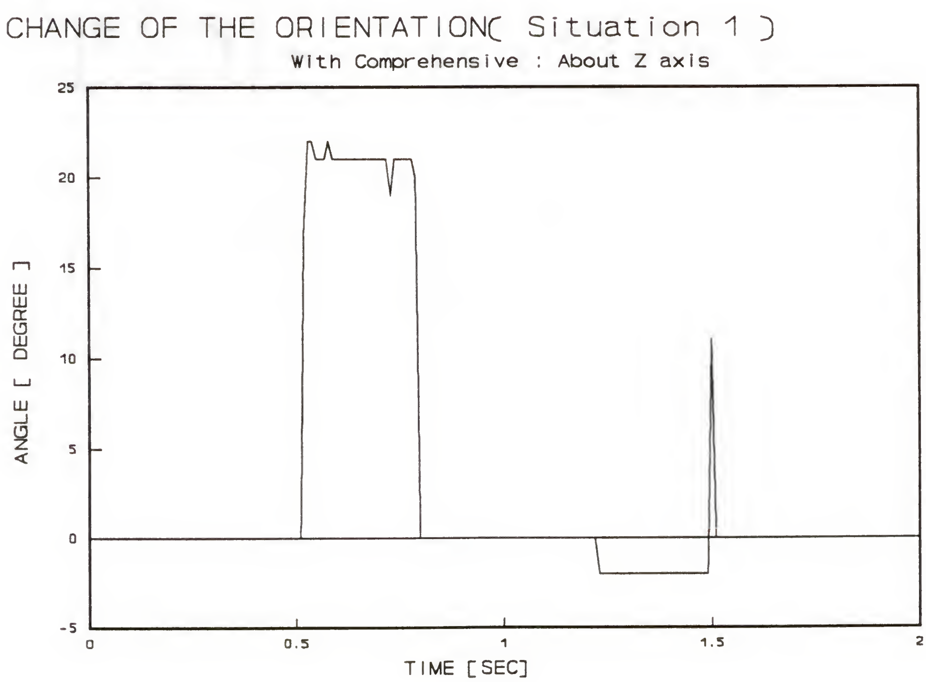

Figure 6.12 Change of The Orientation : About Z axis

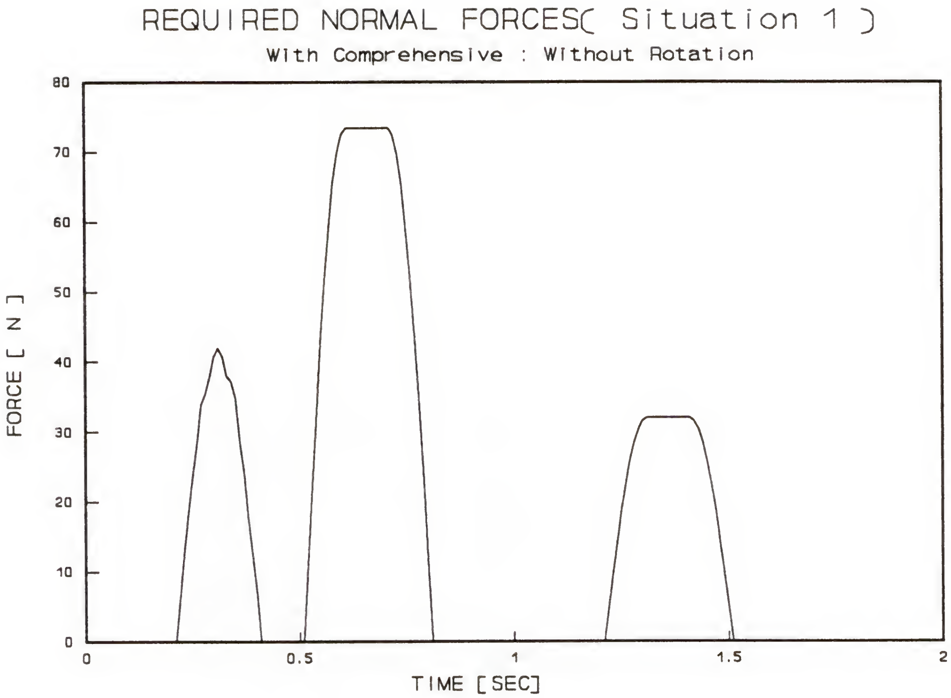

Figure 6.13 Required Normal Forces Without Rotation

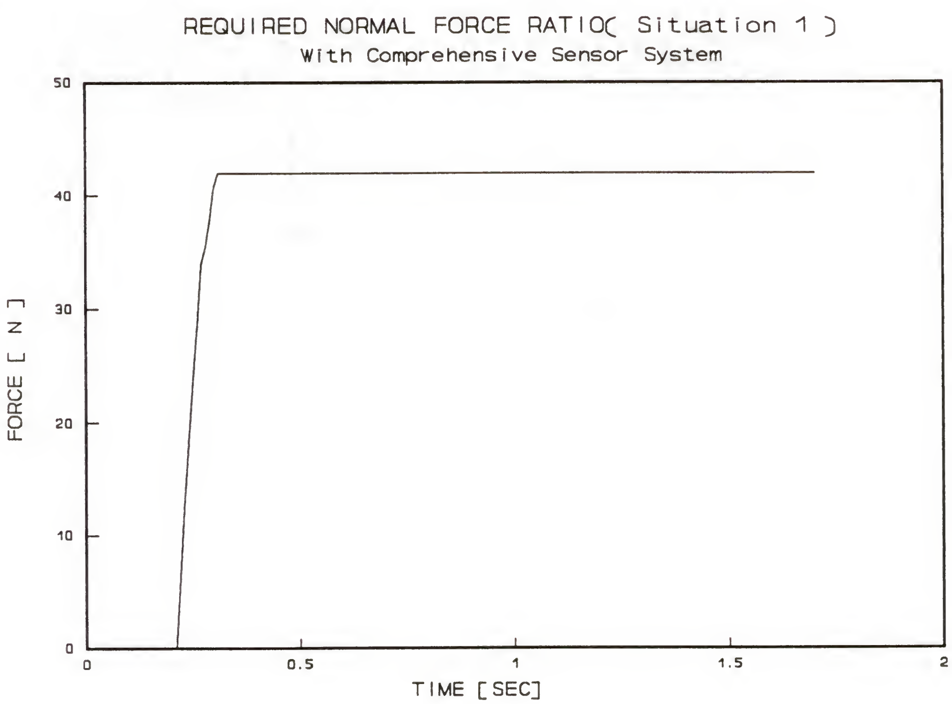

Figure 6.14 Required Normal Force Ratio

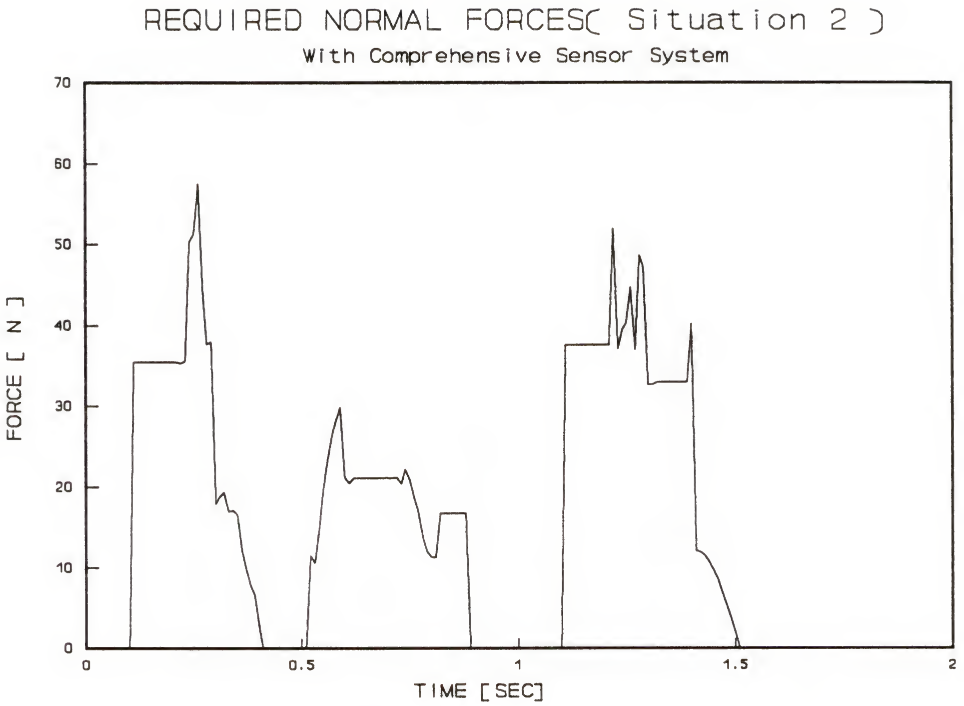

Figure 6.15 Required Normal Forces

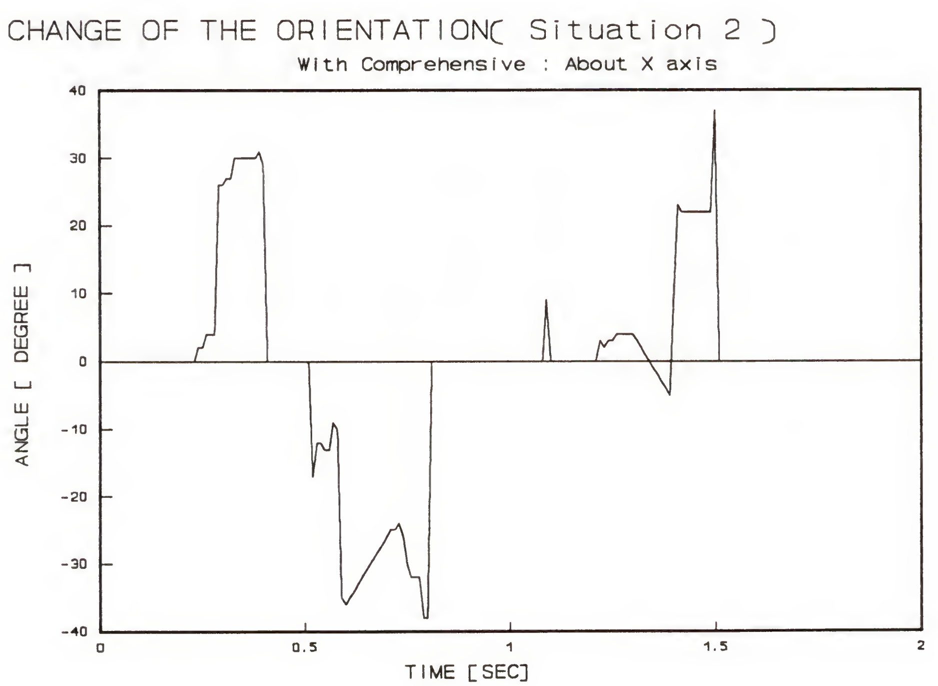

Figure 6.16 Change of The Orientation : About X axis

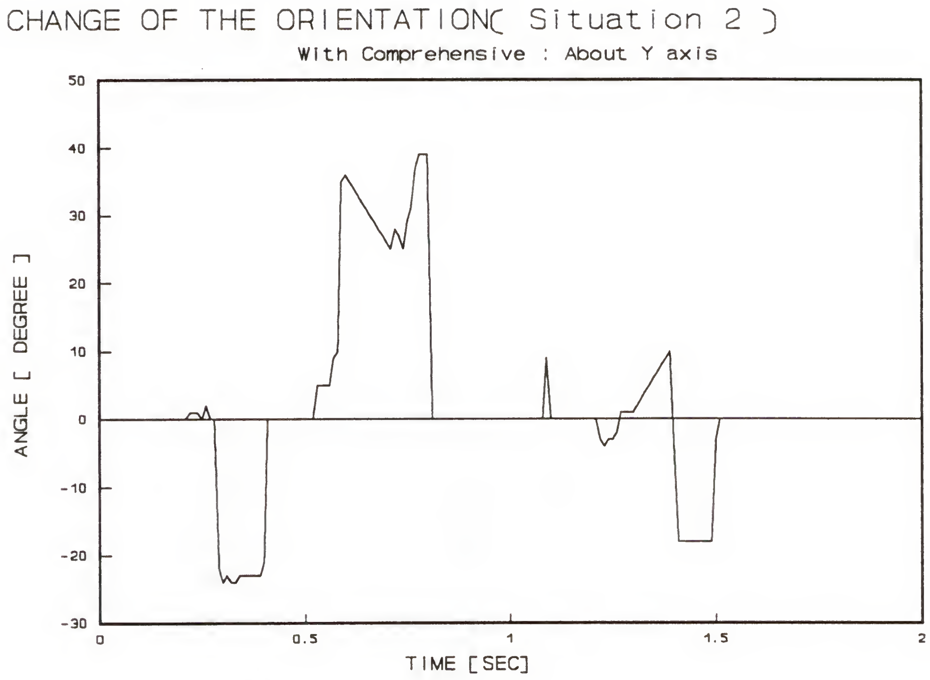

Figure 6.17 Change of The Orientation : About Y axis

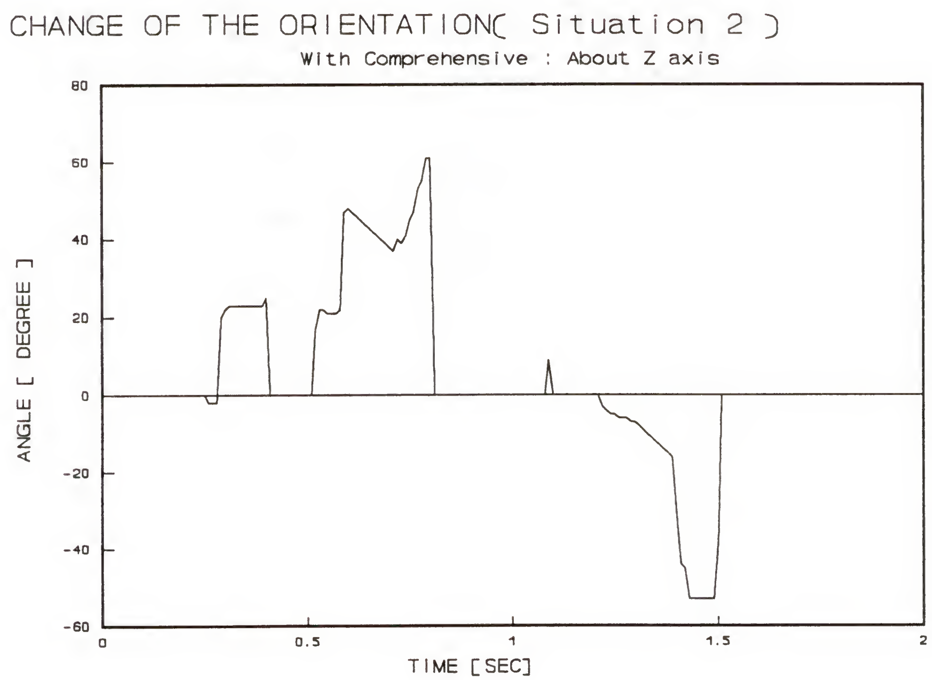

Figure 6.18 Change of The Orientation : About Z axis

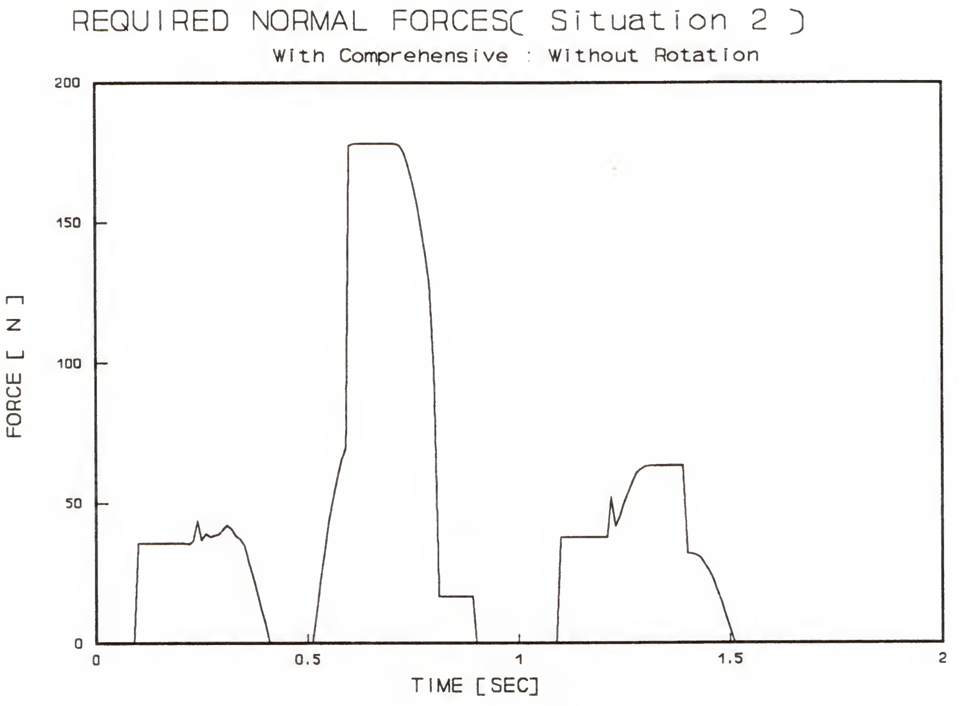

Figure 6.19 Required Normal Forces Without Rotation

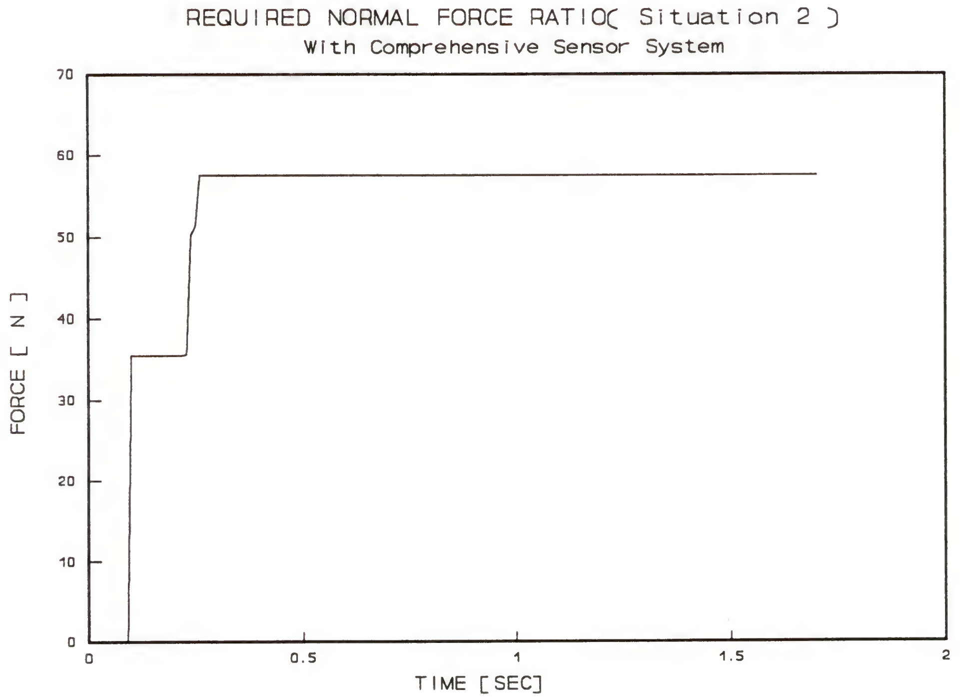

Figure 6.20 Required Normal Force Ratio

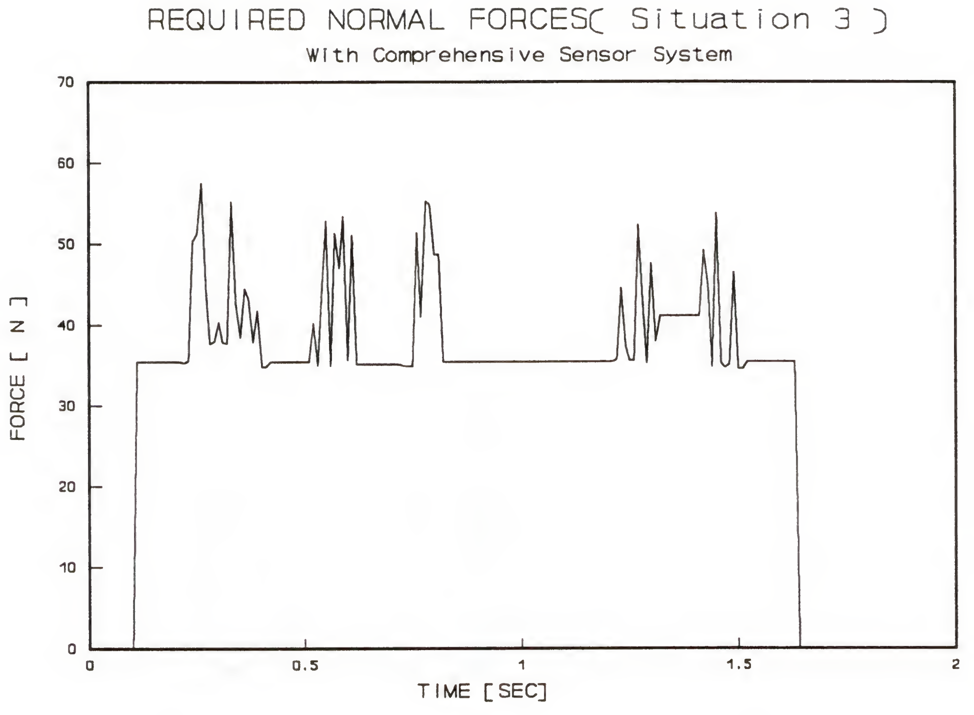

Figure 6.21 Required Normal Forces

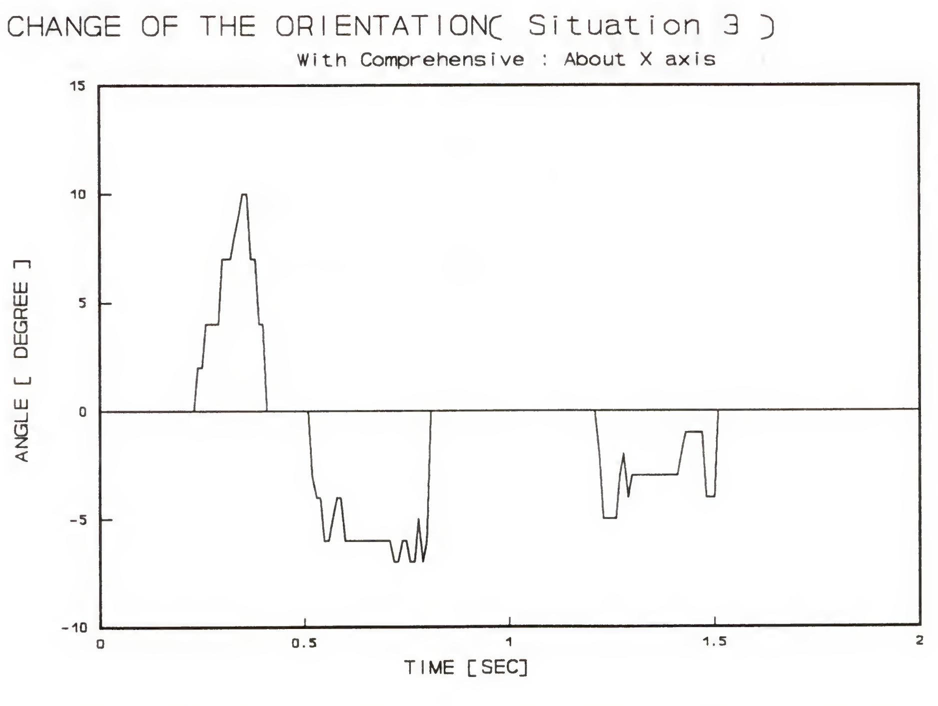

Figure 6.22 Change of The Orientation : About X axis

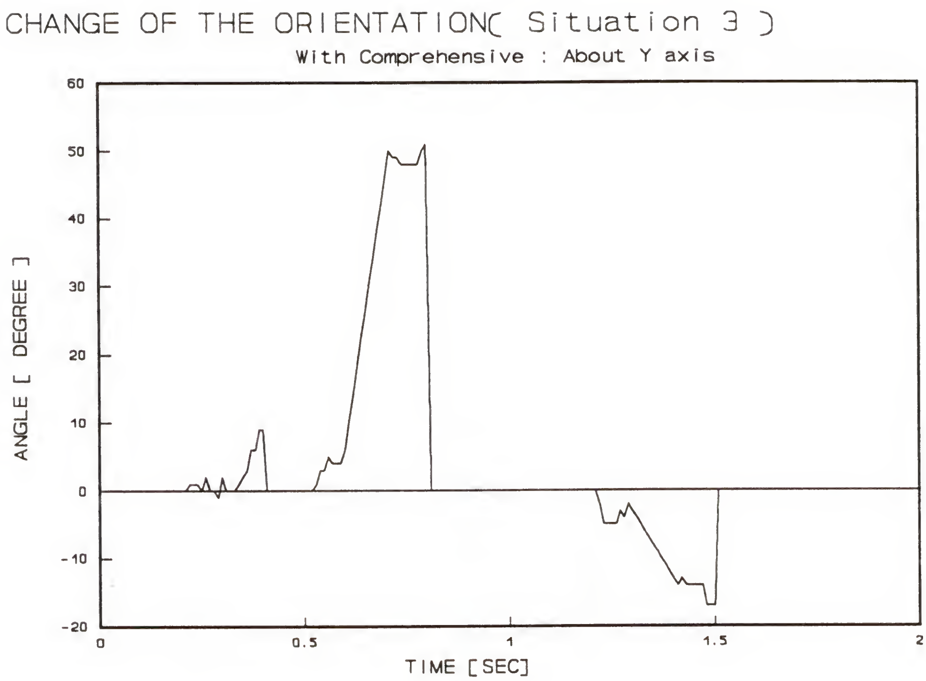

Figure 6.23 Change of The Orientation : About Y axis

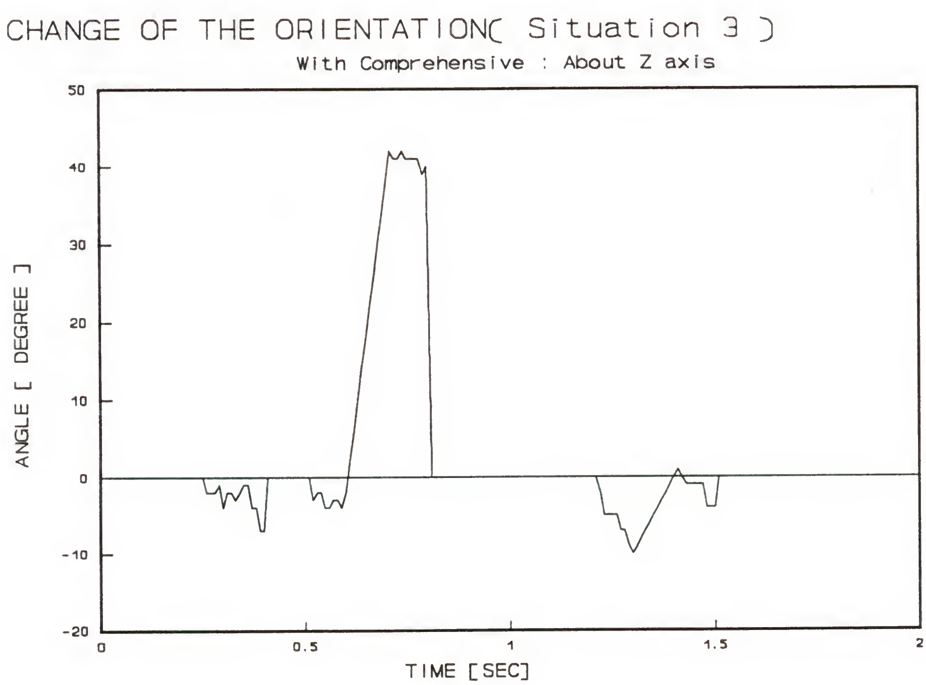

Figure 6.24 Change of The Orientation : About Z axis

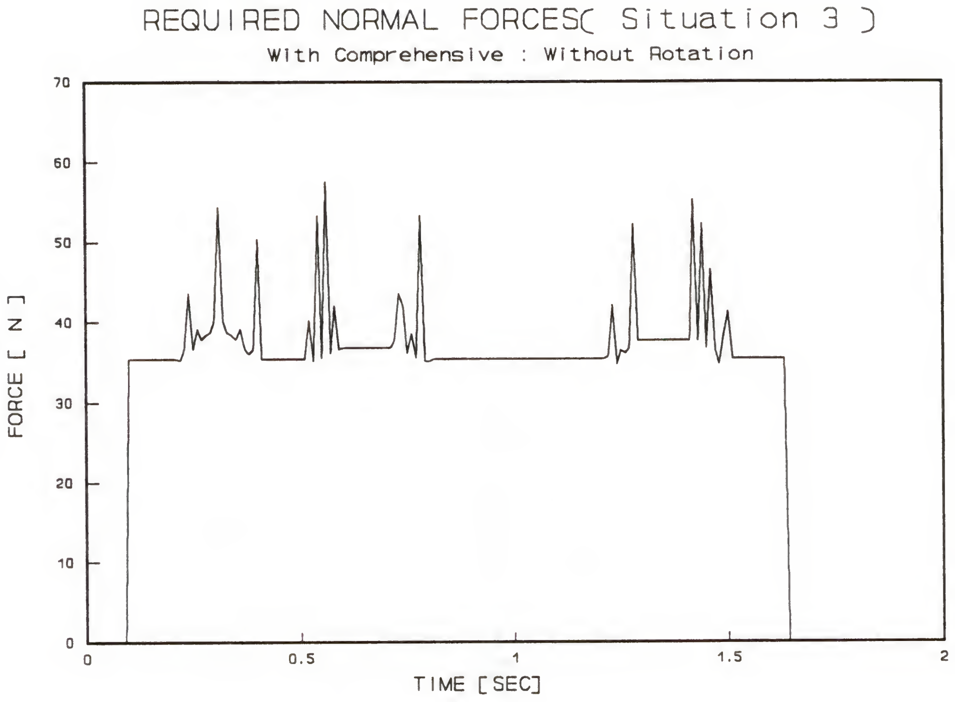

Figure 6.25 Required Normal Forces Without Rotation

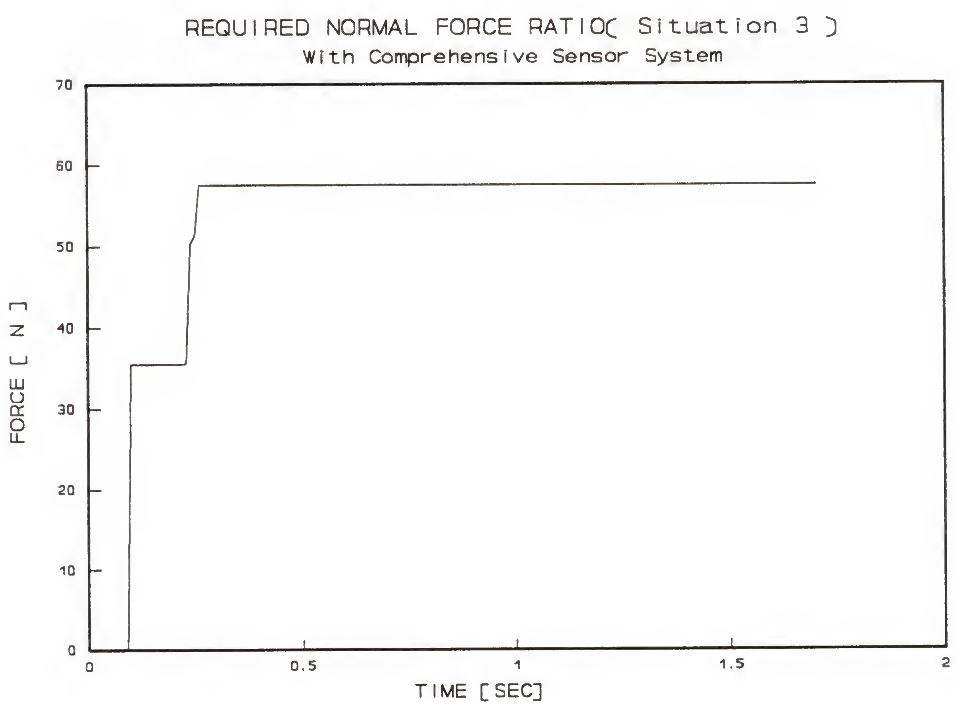

Figure 6.26 Required Normal Force Ratio

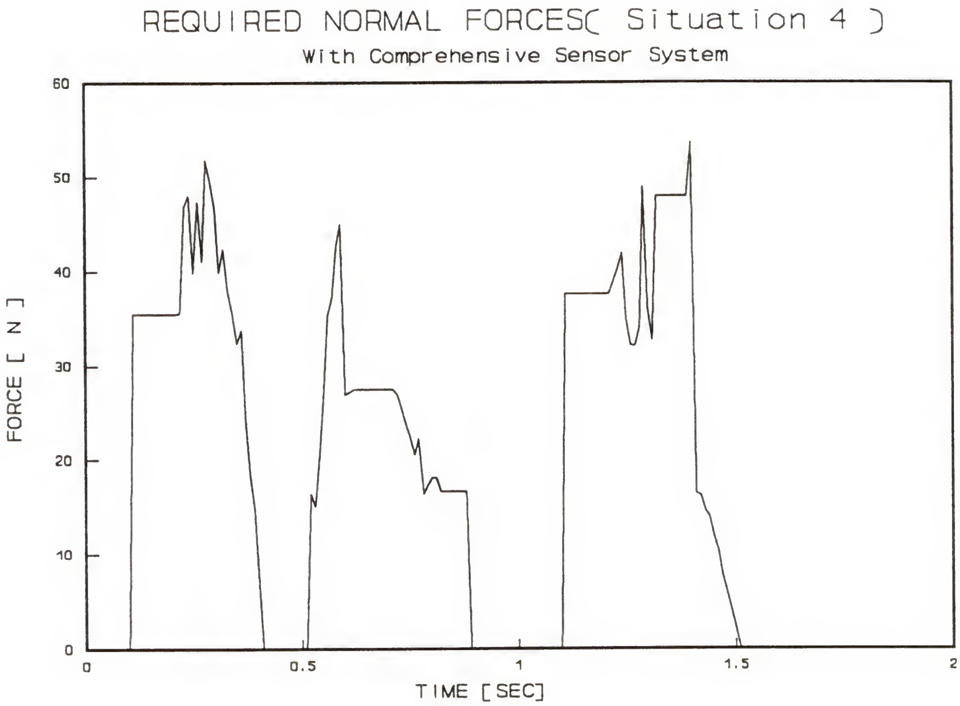

Figure 6.27 Required Normal Forces

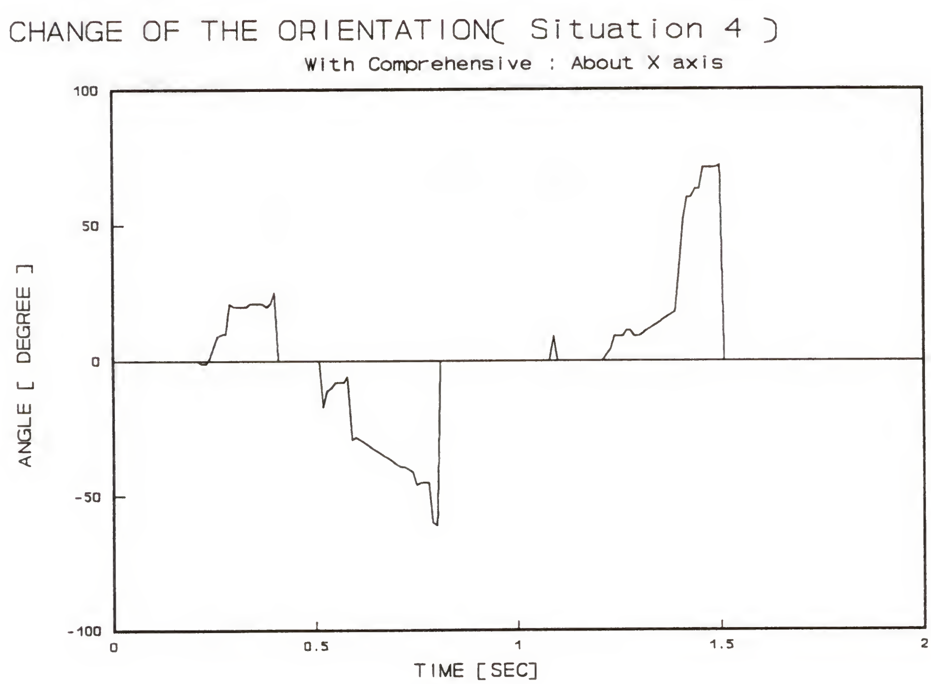

Figure 6.28 Change of The Orientation : About X axis

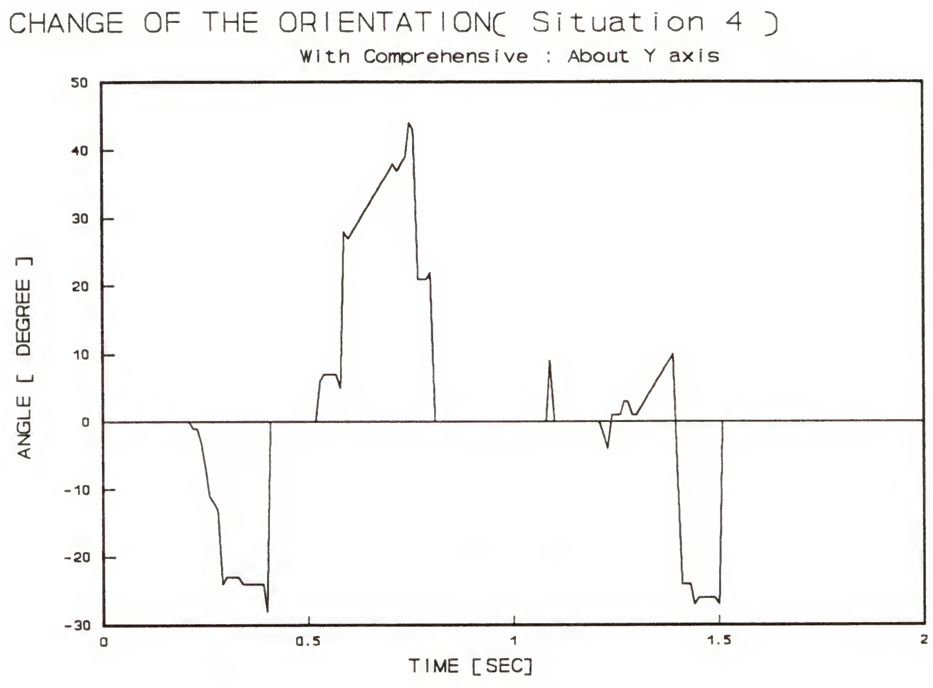

Figure 6.29 Change of The Orientation : About Y axis

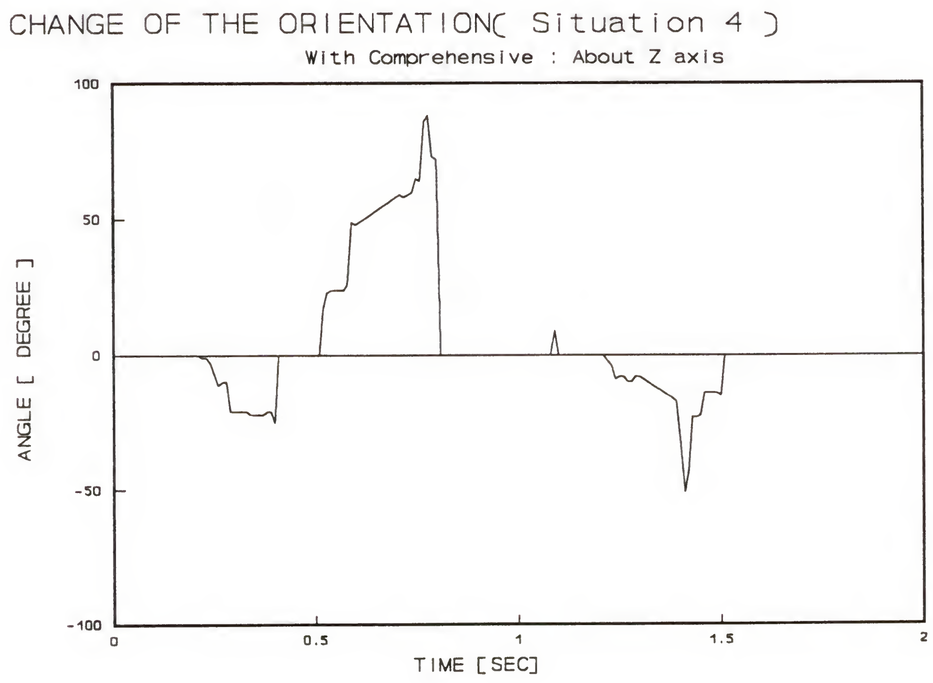

Figure 6.30 Change of The Orientation : About Z axis

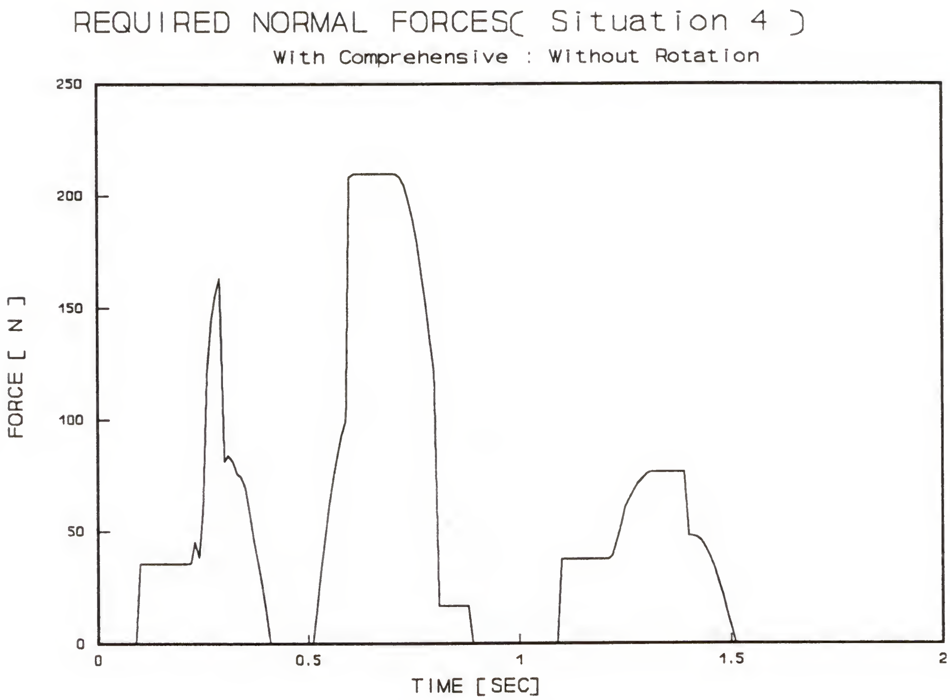

Figure 6.31 Required Normal Forces Without Rotation

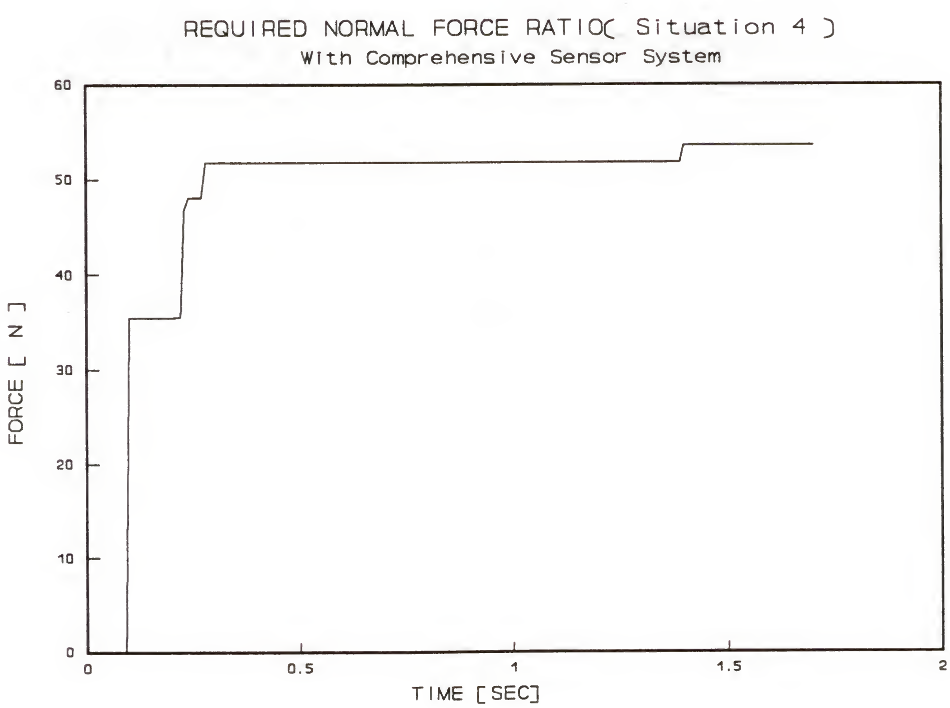

Figure 6.32 Required Normal Force Ratio

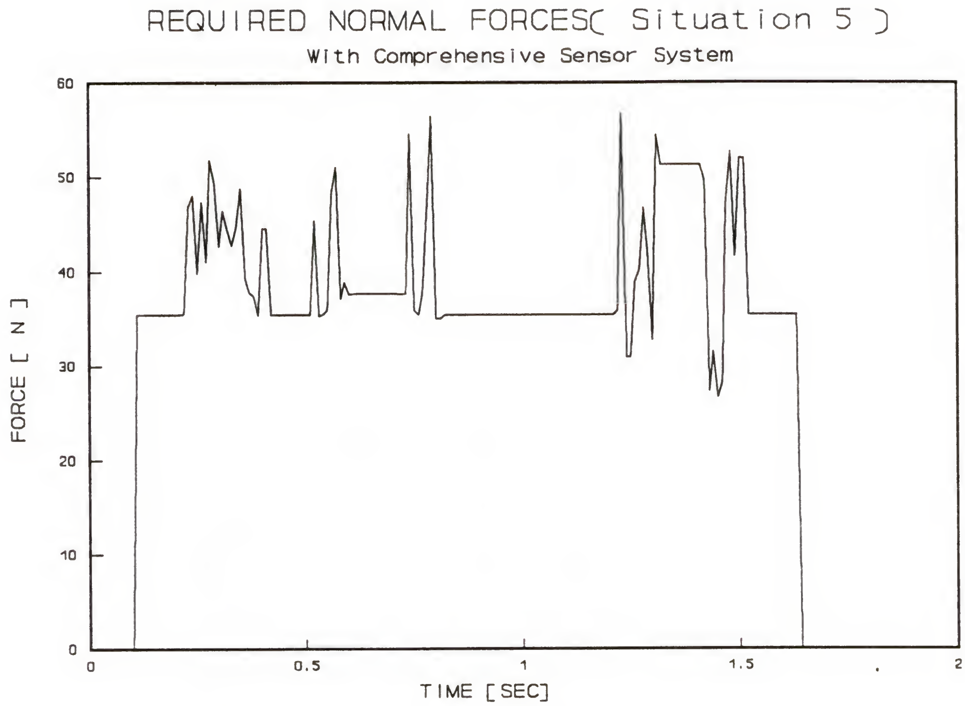

Figure 6.33 Required Normal Forces

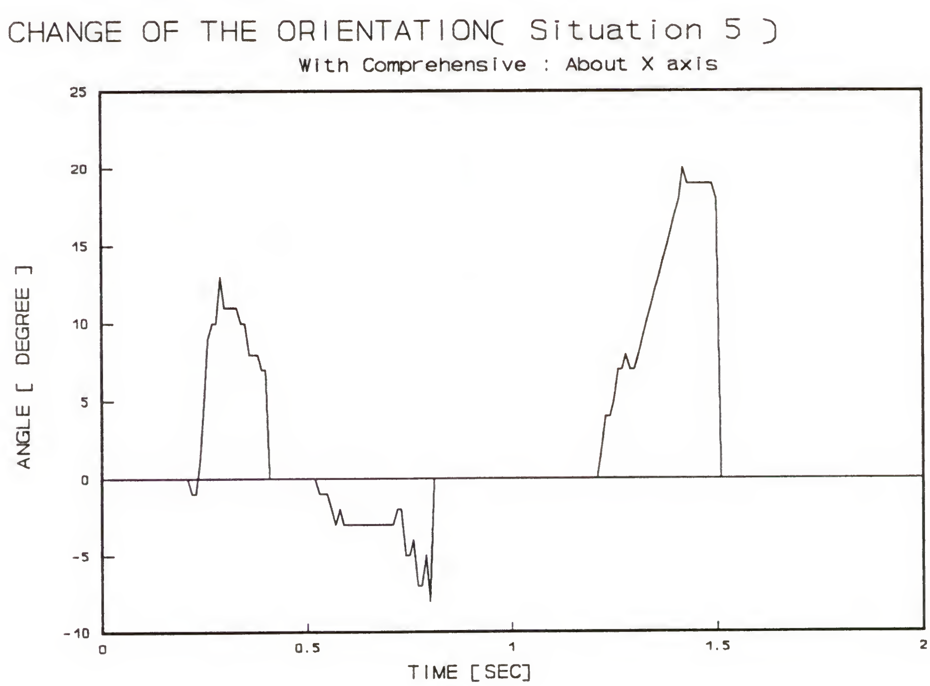

Figure 6.34 Change of The Orientation : About X axis

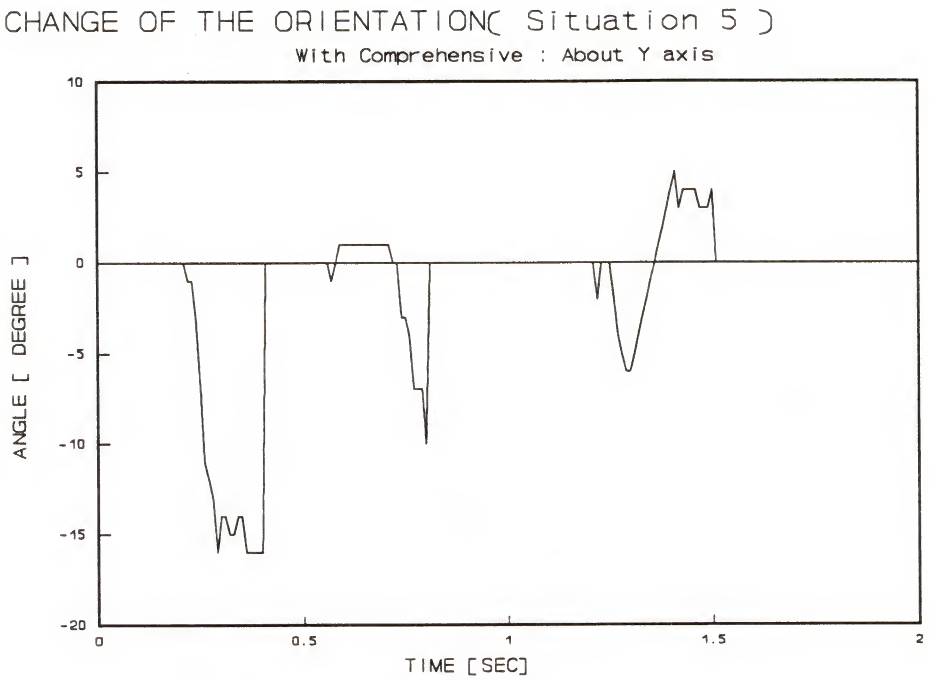

Figure 6.35 Change of The Orientation : About Y axis

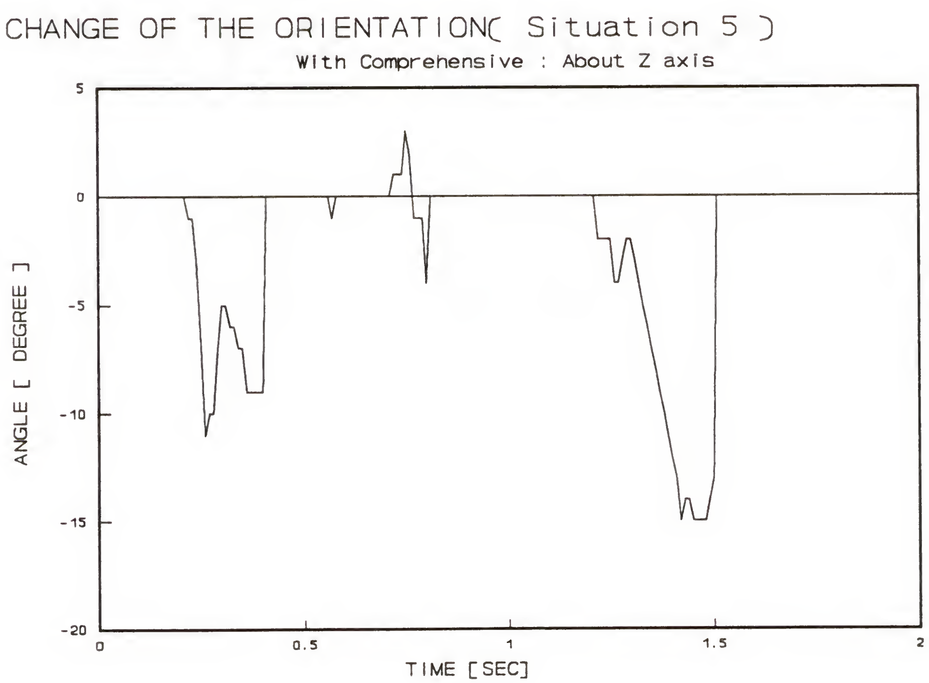

Figure 6.36 Change of The Orientation : About Z axis

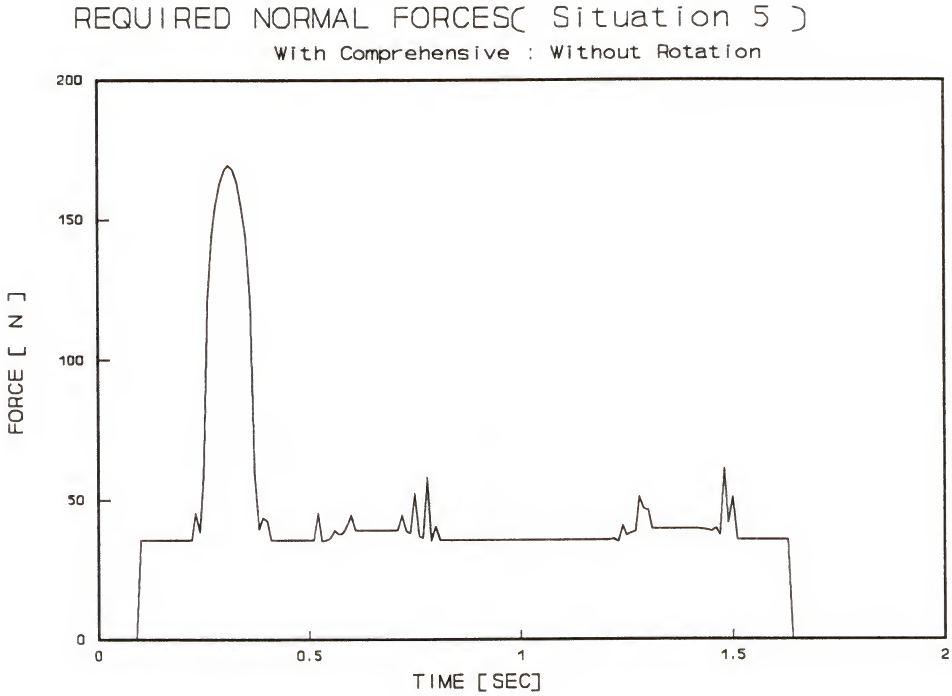

Figure 6.37 Required Normal Forces Without Rotation

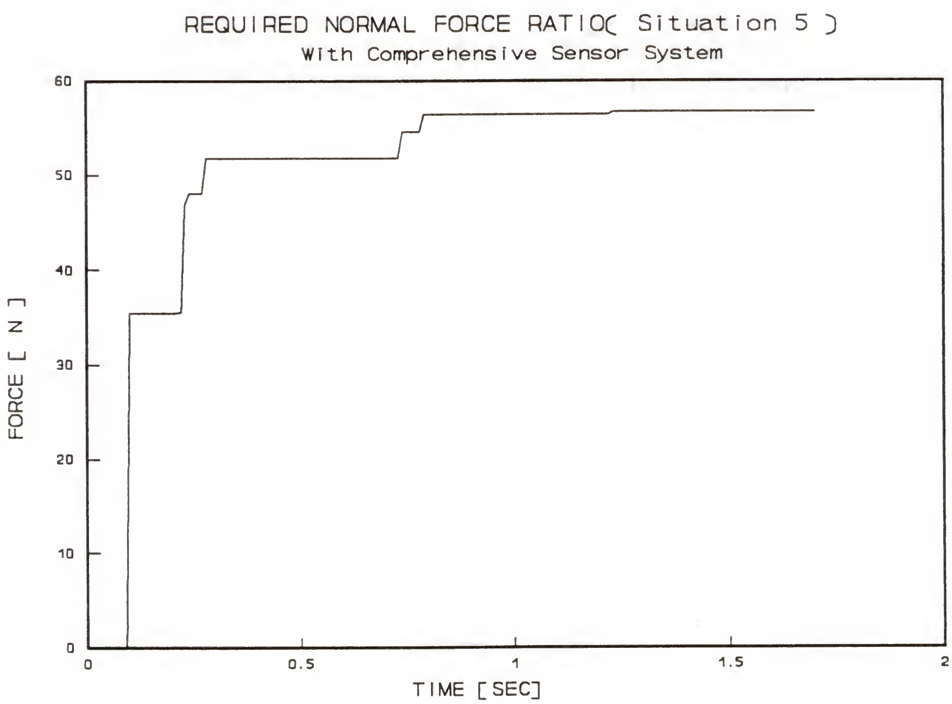

Figure 6.38 Required Normal Force Ratio

6.3 Response With Normal Force Sensors only

Situation 1 (See Figure 6.39 to 6.42)

Because the system is not able to evaluate the exact disturbance, the produced required normal forces are higher in this case than in the cases of where a comprehensive sensor system is used. The changing rate of the orientation is slightly unstable especially when the unexpected force pulse is induced. However, when the moment pulse is induced, the system responses are very stable. The changes of the orientation are relatively small and the change of the required normal forces are also relatively small.

Situation 2 (See Figure 6.43 to 6.46)

Since the disturbances are relatively small in this case, the system can sustain the object with a required normal forces less than 70 N. The changing rates of the orientation also show reasonable ranges during the operation.

Situation 3 (See Figure 6.47 to 6.50)

Since the system has normal force sensors only, it is expected that the required normal forces will be much higher than the system which has a comprehensive sensor system. However, the results for this case are as good as the results with the comprehensive system. From these results it can be stated that the sensor system which has normal force sensors only is satisfactory when the unexpected disturbances are relatively small.

Situation 4 (See Figure 6.51 to 6.54)

Since the disturbances are relatively larger than the previous situations, the required normal forces are higher than before but still are in the feasible ranges. The normal forces which can sustain the stability of the object is considerably reduced by the rotating corrections as can be seen in Figure 6.51.

Situation 5 (See Figure 6.55 to 6.58)

Despite the fact that the disturbances are relatively large in this case, the system is very stable when the unexpected disturbances occur in conjunction with some expected disturbance. The changing rate of the orientation is small during the operation and the required normal forces are less than 70 N except for only one instant.

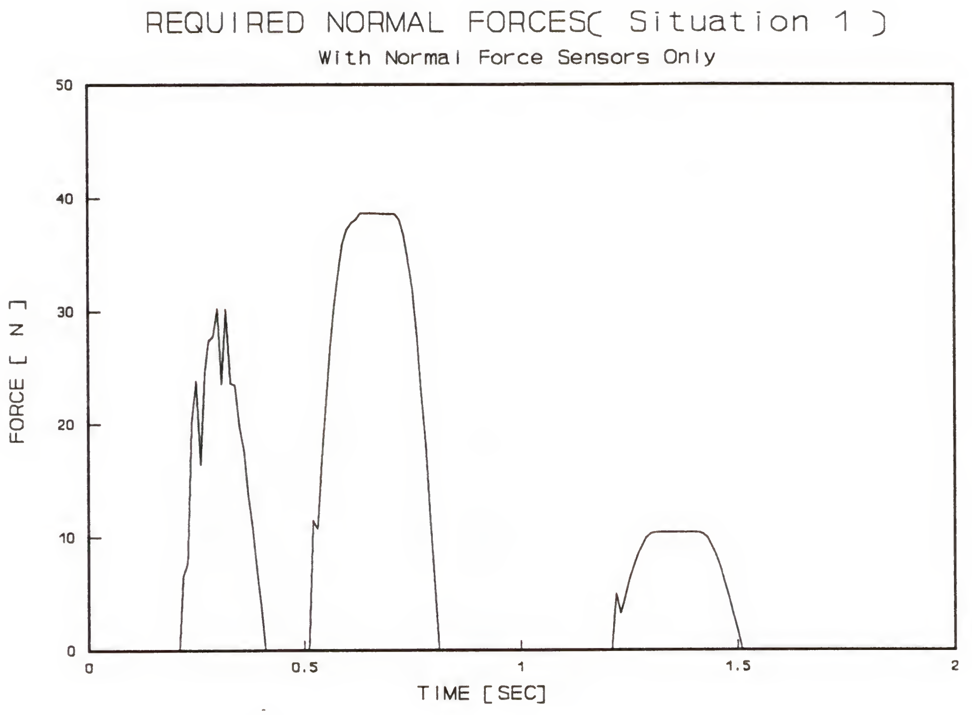

Figure 6.39   Required Normal Forces

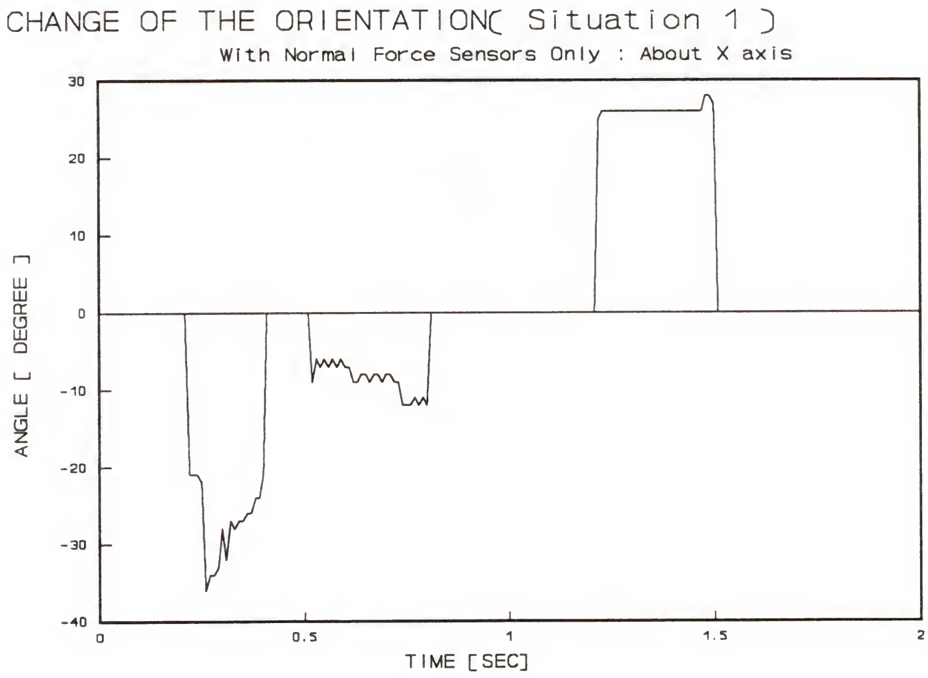

Figure 6.40   Change of The Orientation : About X axis

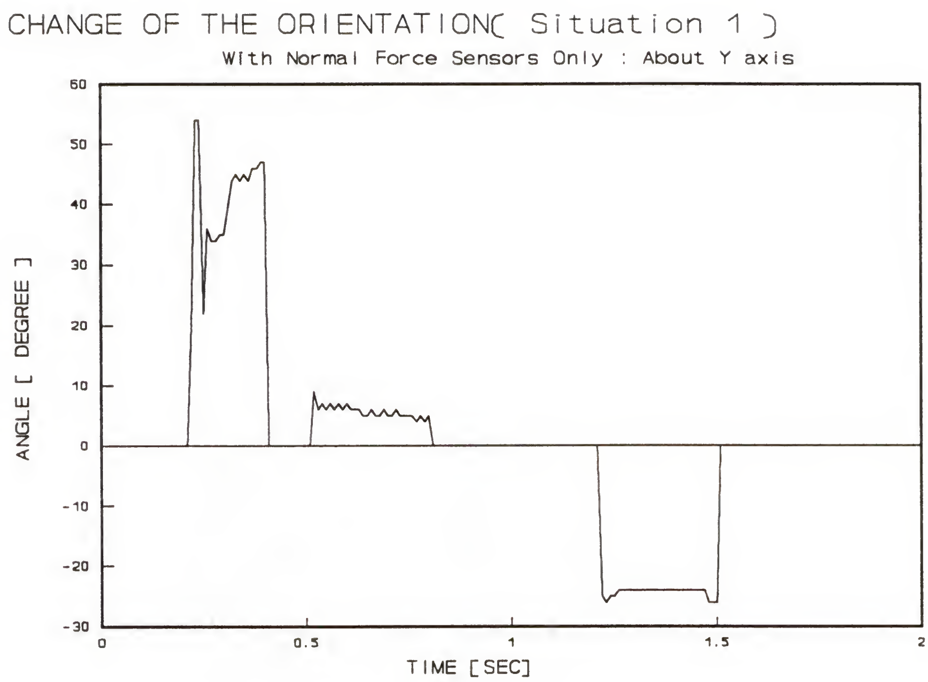

Figure 6.41 Change of The Orientation : About Y axis

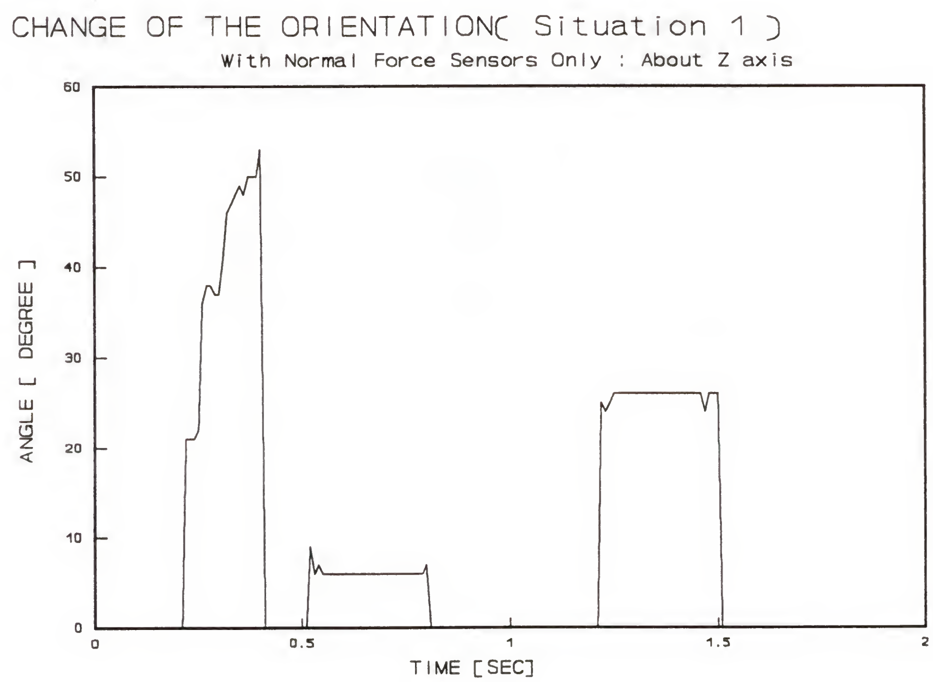

Figure 6.42 Change of The Orientation : About Z axis

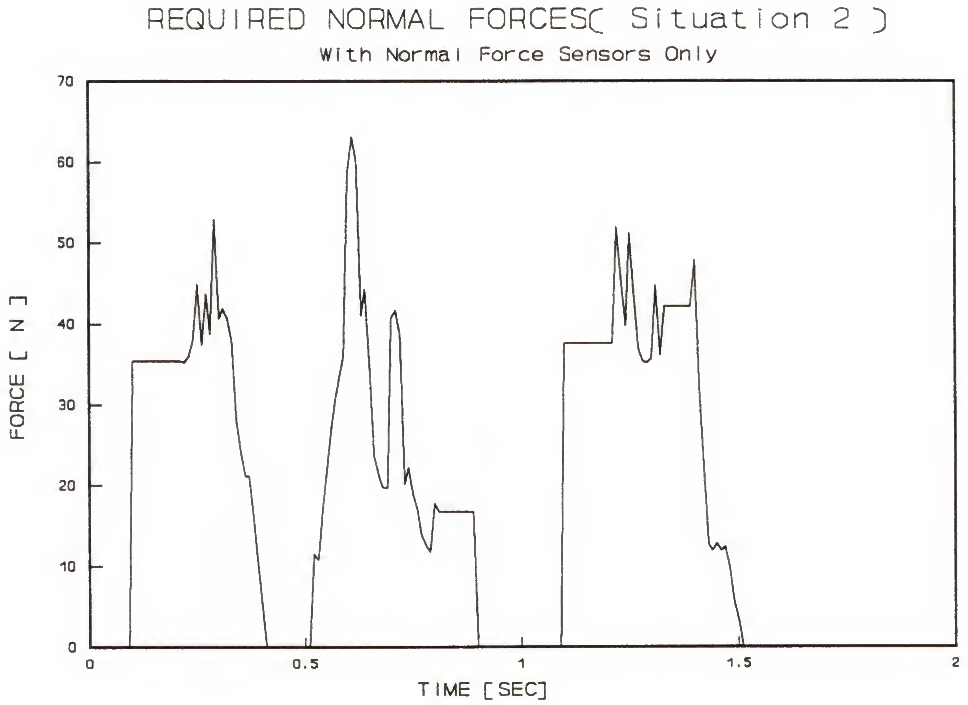

Figure 6.43 Required Normal Forces

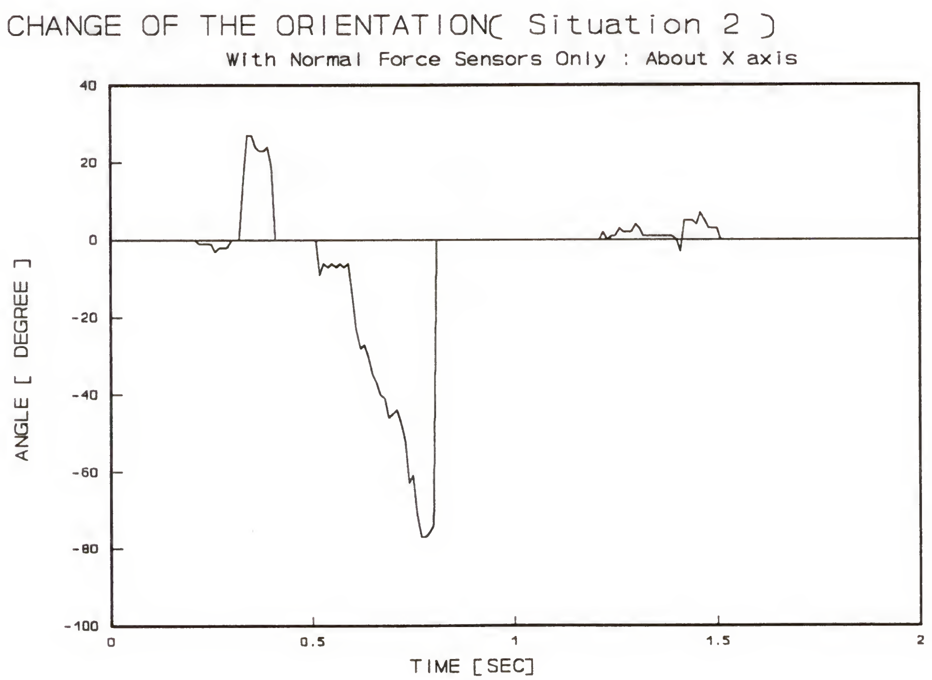

Figure 6.44 Change of The Orientation : About X axis

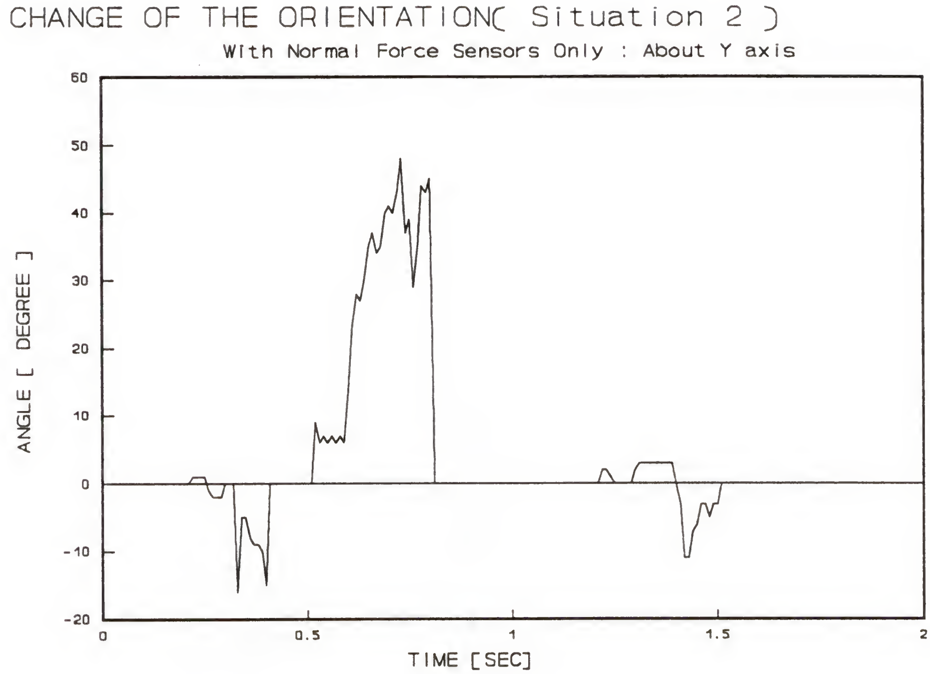

Figure 6.45 Change of The Orientation : About Y axis

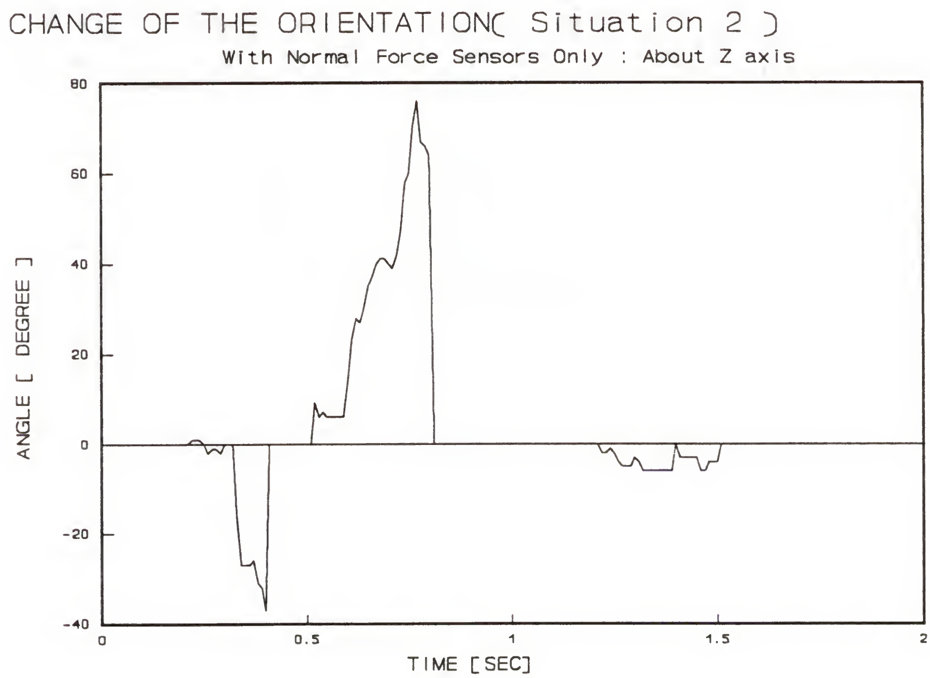

Figure 6.46 Change of The Orientation : About Z axis

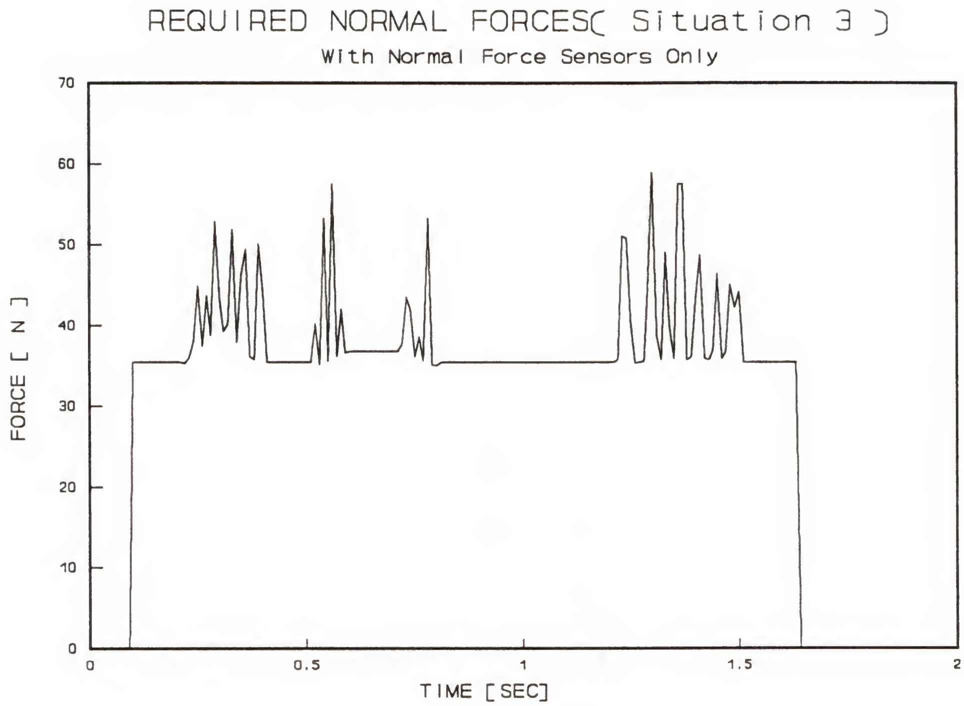

Figure 6.47 Required Normal Forces

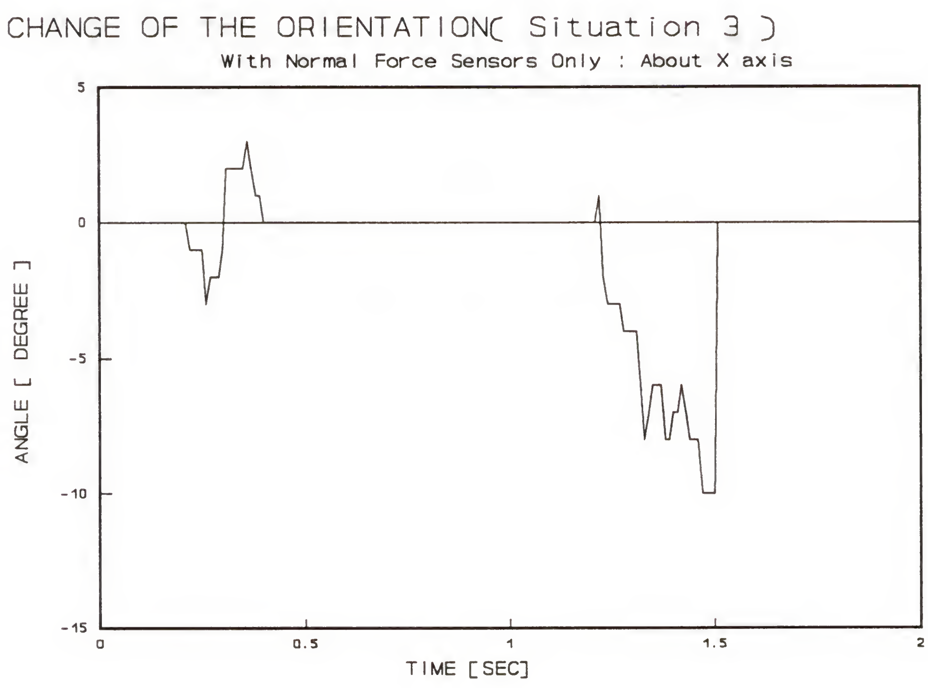

Figure 6.48 Change of The Orientation : About X axis

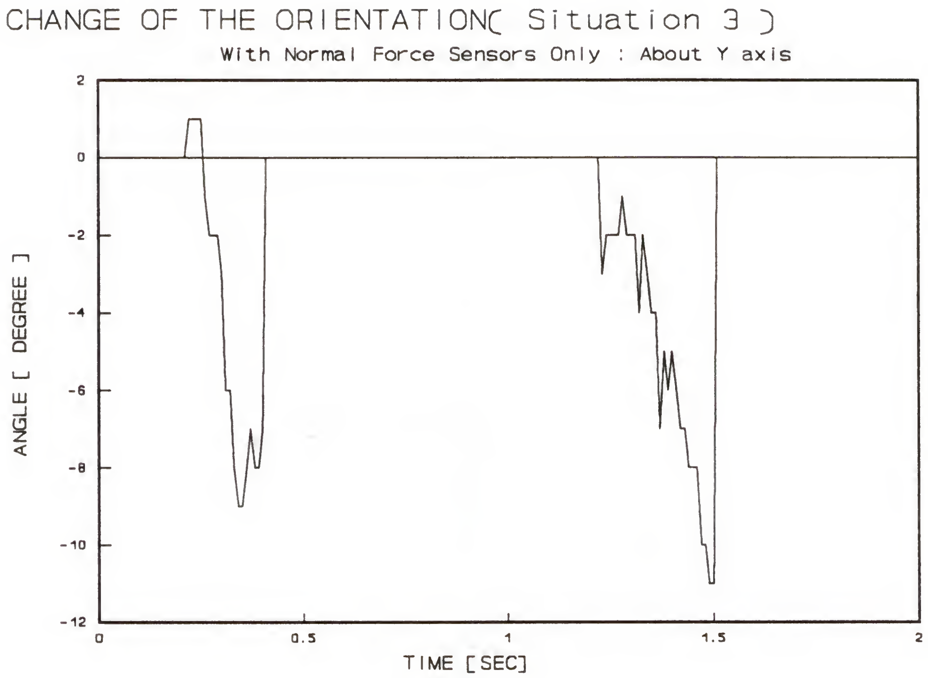

Figure 6.49 Change of The Orientation : About Y axis

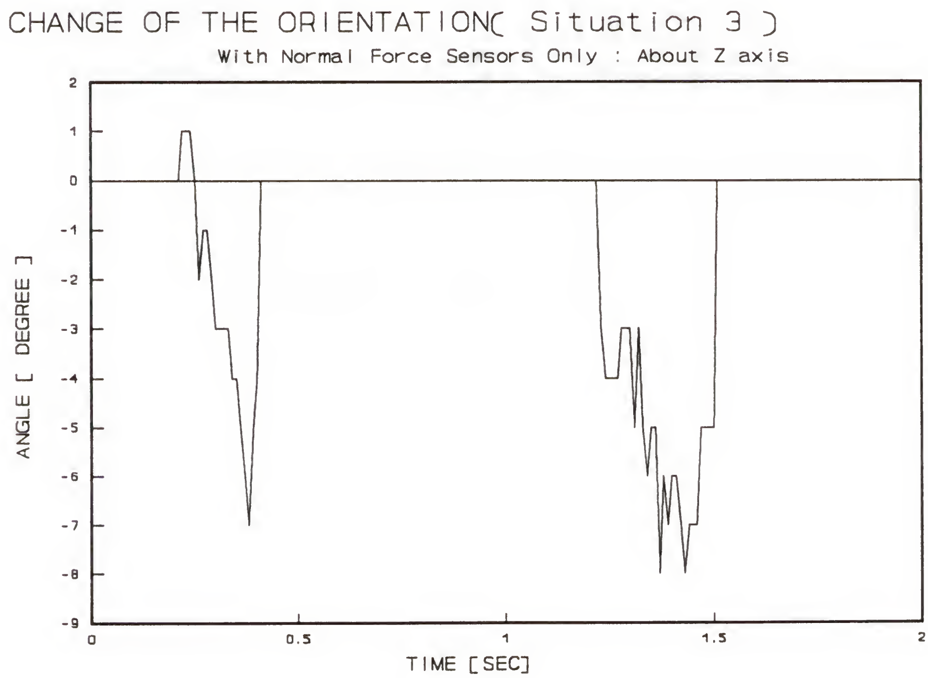

Figure 6.50 Change of The Orientation : About Z axis

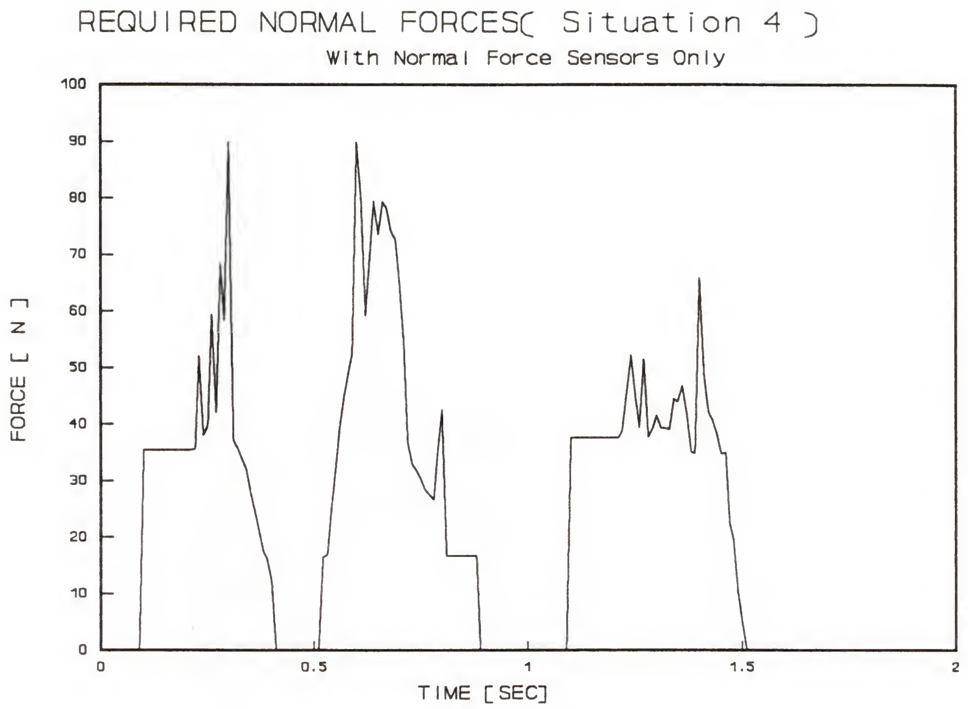

Figure 6.51 Required Normal Forces

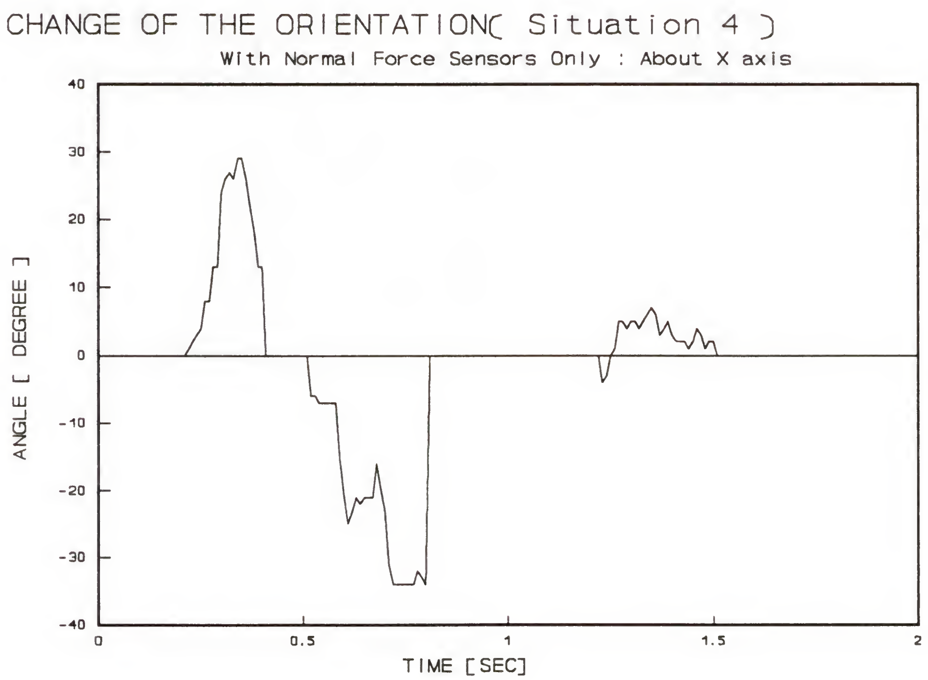

Figure 6.52 Change of The Orientation : About X axis

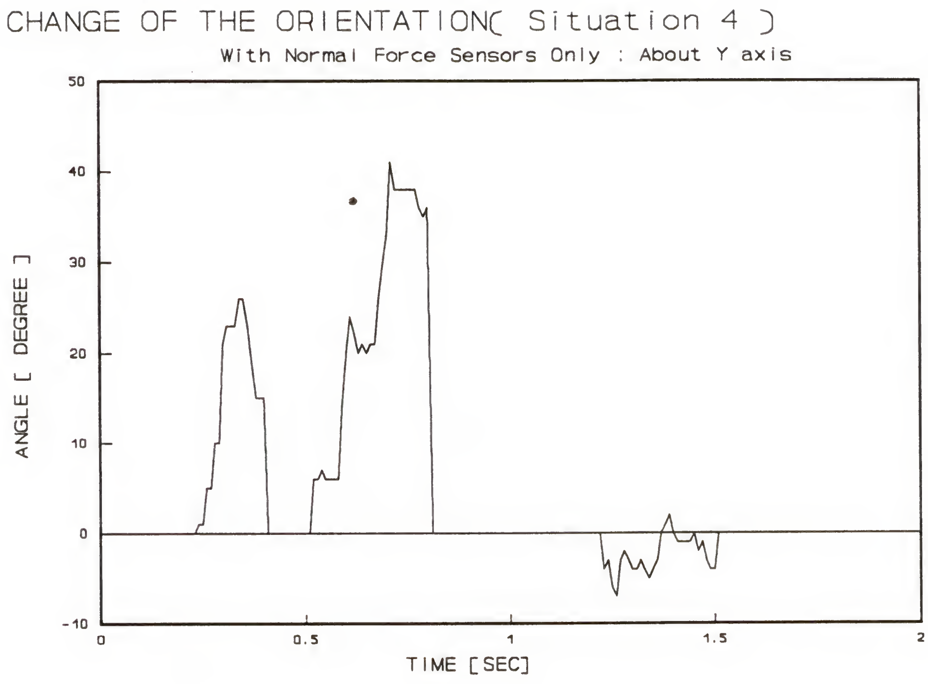

Figure 6.53 Change of The Orientation : About Y axis

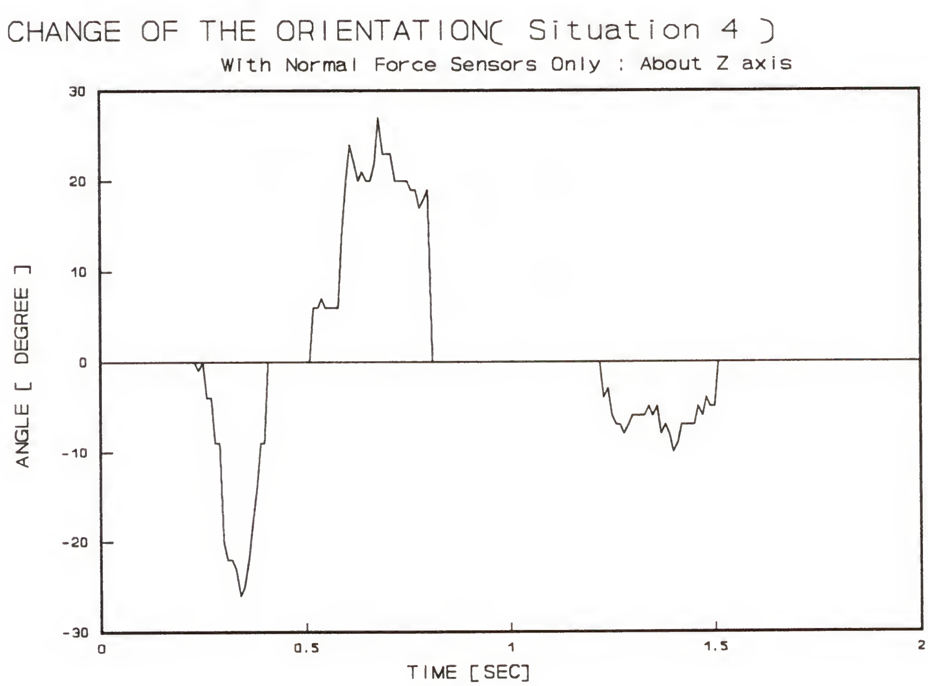

Figure 6.54 Change of The Orientation : About Z axis

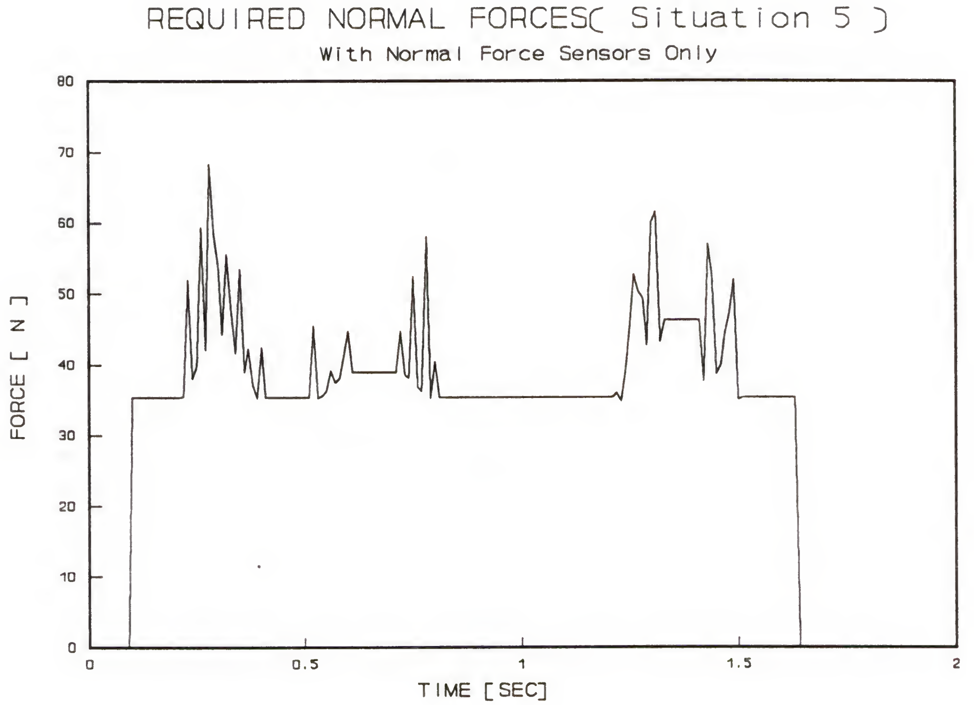

Figure 6.55 Required Normal Forces

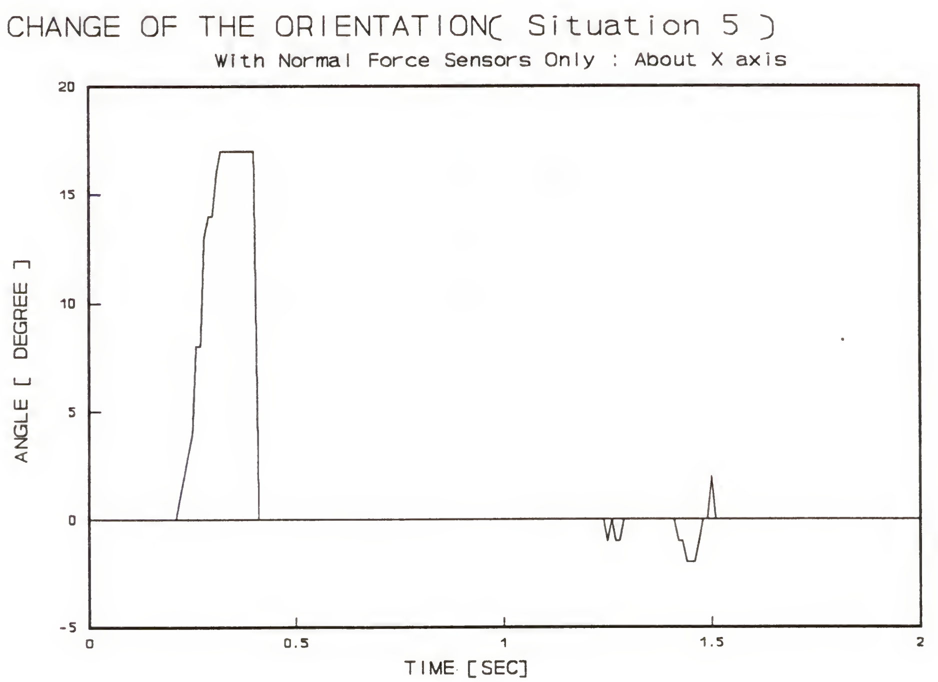

Figure 6.56 Change of The Orientation : About X axis

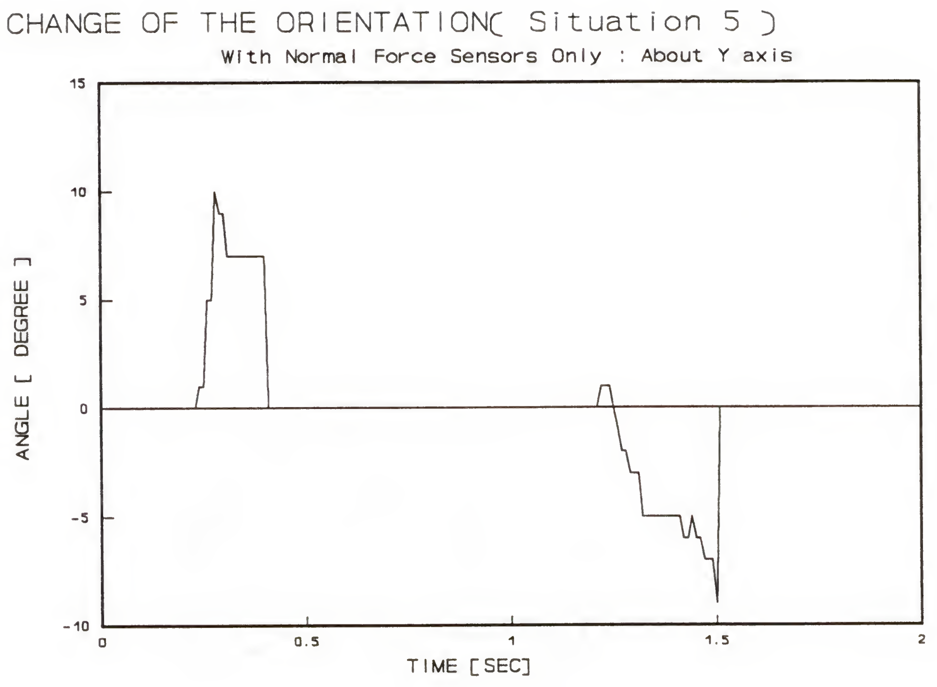

Figure 6.57 Change of The Orientation : About Y axis

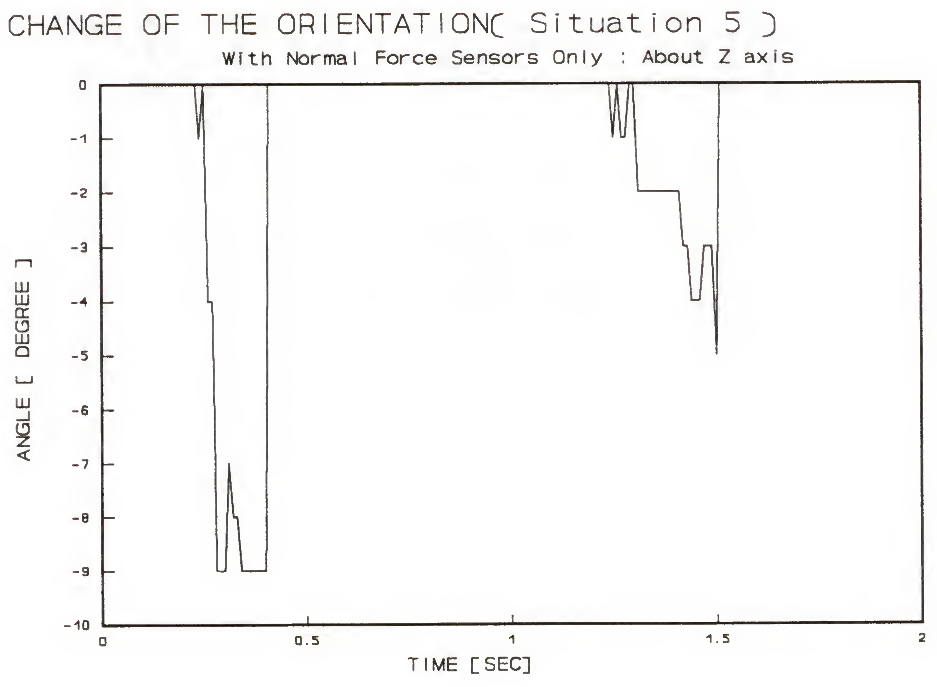

Figure 6.58 Change of The Orientation : About Z axis

6.4 Conclusions

. The system which has normal force sensors only is satisfactory when the moment disturbances occur at a time without any expected disturbance.

. The system with a comprehensive sensor system can always sustain the stability of the object if the unexpected disturbance which is detected at first is resistably at that instant.

. If the object can sustain large normal forces without damage, a system with normal force sensors only can be utilized and higher normal forces will generally be estimated to resist the unexpected disturbances.

. By monitoring the emergency factor which is defined in chapter 4, the manipulator can add the required normal force instantly in a real time control. If the system has a value of the factor less than 1.0 and does not reduce the normal forces once they are increased, rapid change of the normal force can be avoided as illustrated in the Figures.

CHAPTER 7
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary

In the present work, control algorithms for an articulate manipulator to sustain the stability of the grasped object are developed to deal with situations where some unexpected disturbance occur while it performs a preplanned operation. The study reported in this thesis can be summarized as follows.

1. An analysis of a soft contact with friction is developed in chapter 2. In this analysis, ideal independent linear springs and angular springs were considered. The stiffness of the virtual spring system which combines these two ideal springs, and the relationship between the moment and the normal force at the contact were derived.

2. A procedure for finding the optimal configuration of the grasp based on the previous analysis is presented in chapter 3. The optimal grip is also characterized by a gripping ratio which is the required normal force ratio between the fingers within the friction constraints when the gripper grasps an object in its initial weightless equilibrium

state . Since the end effector does not produce any net force on the object , it is always in an equilibrium state whether or not there is an external force. This can be determined by a linear programming optimization in a very short (c.p.u)time. If the manipulator is to perform a preplanned operation, the optimal configuration which requires the minimum normal forces at the fingertips can be found by an optimization procedure considering all the expected force and moment through out the movement including any inertial effects. The manipulator can perform the preplanned operation successfully by this optimal configuration which is found before starting the operation if there is no unexpected disturbance during the operation.

3. In order to deal with any unexpected disturbance, identification of the disturbance is needed first to resist it effectively. This identification process with an optimal grasp is discussed in chapter 4. Several computational algorithms are introduced based on two types of sensor systems. One is a comprehensive sensor system which can sense the direction and magnitude of all the frictional and normal forces at the fingertips. The other sensor system utilizes normal force transducers only at the fingertips. An extrapolation procedure is introduced to anticipate the next change of the disturbance when the comprehensive sensor system is used. A Heuristic approach utilizing the information from the system which has normal force sensors only is introduced.

4. Several control algorithms which can sustain the stability of the object when an unexpected disturbance is detected during the operation are developed in chapter 5. The proposed corrective actions are implemented by changing the required normal forces and changing the orientation of the object. These algorithms are relatively simple and are easy to incorporate in a real time control. The algorithms are developed with consideration of all the possible events which may occur during the operation.

5. To examine the feasibility and limitations of these control algorithms, some illustrative computer simulations were performed for several selected circumstances and the results are presented in chapter 6.

7.2 Conclusion

The following can be concluded from the results of this study.

1. If the manipulator has a comprehensive sensor system as described in chapter 5, it can perform a preplanned operation successfully by the control algorithms developed in this study if the unexpected disturbances are manageable within the limits imposed on the maximum normal forces.

2. If the manipulator has normal force sensors only, the required normal forces are usually higher in comparison with the case where a comprehensive sensor system is used.

3. If the end effector has normal force sensors only, the approach proposed in chapter 5 for utilizing the sensed normal force information for the next step during the identification appears to be feasible in the selected examples.

4. The computation time(c.p.u time) is one of the critical factors in the control system. If it is necessary to rotate the object more than 10 times in the unit degree about its axes, the computational time would be more than 50 % of the 0.01 sec time needed for data acquisition in the computer simulation.

5. In the geometric analysis of the manipulators arm movement, the computation is assumed to be completed within the data acquisition time and the actuators are assumed to respond fast enough to perform the task of changing the orientations.

7.3 Recommendations

The followings are recommendations for future investigations.

1. Special attention should be given to minimizing the computation time by the design of a dedicated micro processor which is optimized to meet the special requirement of the proposed system.

2. The reaction time of the integrated sensor - control - actuator system should be minimized in order to meet the demanding requirements of responding to fast occurring unexpected disturbances.

APPENDIX A
TRANSFORMATION MATRICES

Given a force and moment vector acting at the origin of some coordinate frame attached to a fixed object, the objective is to find force and moment vector acting at some other coordinate frame which is also attached rigidly to the object. A generalized transformation matrix can be derived for transforming forces and moments between the coordinate frames when the position vector between the two frames is known.

A transformation matrix for a contact point is defined as;

$$T_i = \begin{bmatrix} [R] & [0] \\ [\tilde{P}\,] & [R] \end{bmatrix}_i \tag{A.1}$$

$$R = \begin{bmatrix} l_x & m_x & n_x \\ l_y & m_y & n_y \\ l_z & m_z & n_z \end{bmatrix}, \qquad \tilde{P} = \begin{bmatrix} l_z y - l_y z & m_z y - m_y z & n_z y - n_y z \\ l_x z - l_z x & m_x z - m_z x & n_x z - n_z x \\ l_y x - l_x y & m_y x - m_x y & n_y x - n_x y \end{bmatrix} \tag{A.2}$$

Where R is a rotation matrix. The given position vector is $[\, x \,,\, y \,,\, z \,]^t$. As an example in order to derive the transformation matrix from frame B to frame C(Figure A.1), $F_b = [\ 10,\ 0,\ 0,\ 0,\ 100,\ 0\ ]^t$ is the force and moment

vector acting at the origin of frame B, the equivalent force and moment vector in frame C can be formulated as follows;

If $R = \lfloor R_{x_{90}} \rfloor \cdot \lfloor R_{y_{90}} \rfloor$, position vector $P = [\ 10,\ 5,\ 10\ ]^t$ then

$F_c = T \cdot F_b = [\ 0,\ 0,\ 10,\ 100,\ 50,\ 0\ ]^t$.

From these transformation matrices, constraints matrices can be produced as described in chapter 3.

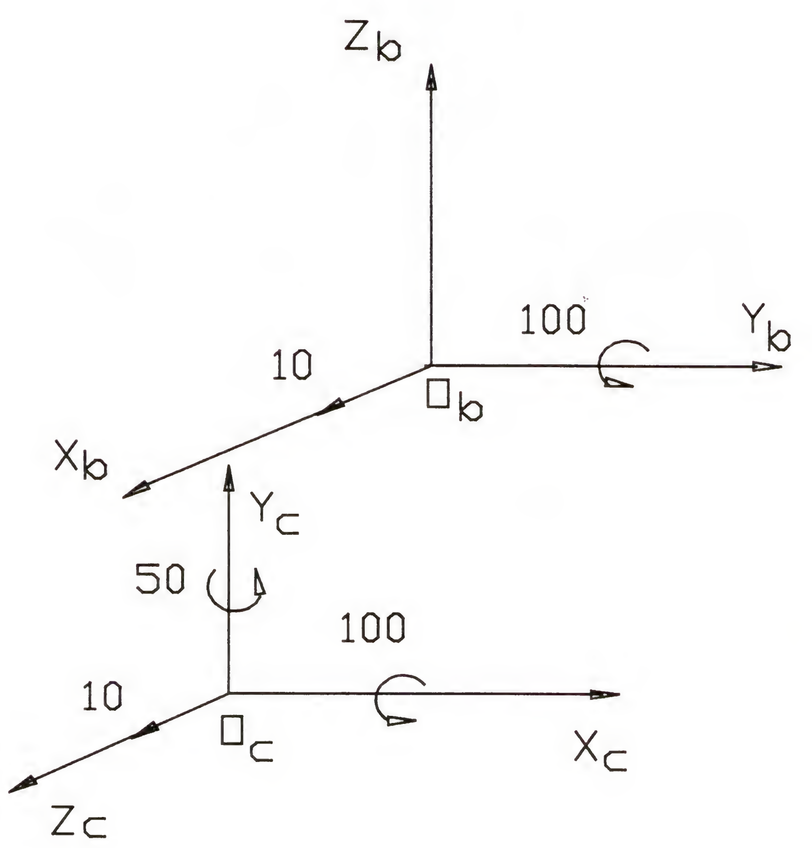

Figure A.1 Equivalent Force and Moment

APPENDIX B
POLYNOMIAL APPROXIMATION

The third-order approximating polynomial can be written as :

$$f = a_o + a_1 X + a_2 X^2 + a_3 X^3 \quad \text{(B.1)}$$

In the following equations for the coefficients a_0 , a_1 , a_2 and a_3 ,it should be remembered that the ordering of the point X_1 , X_2 , X_3 and X_4 is arbitrary so that, for example, X_3 may be less than X_1. Also, it is not required that the points be equally spaced.

Two-point linear approximation

Given ; (X_1, f_1), (X_2, f_2)

$$a_3 = 0 \quad \text{(B.2)}$$

$$a_2 = 0 \quad \text{(B.3)}$$

$$a_1 = \frac{f_2 - f_1}{X_2 - X_1} \quad \text{(B.4)}$$

$$a_0 = f_1 - a_1 X_1 \quad \text{(B.5)}$$

<u>Three-point Quadratic approximation</u>

Given ; (X_1, f_1),(X_2, f_2),(X_3 , f_3)

$$a_3 = 0 \tag{B.6}$$

$$a_2 = \frac{\dfrac{(f_3 - f_1)}{(X_3 - X_1)} - \dfrac{(f_2 - f_1)}{(X_2 - x_1)}}{X_3 - X_2} \tag{B.7}$$

$$a_1 = \frac{f_2 - f_1}{X_2 - X_1} - a_2 \ (X_1 + X_2) \tag{B.8}$$

$$a_0 = f_1 - a_1 X_1 - a_2 X_1^2 \tag{B.9}$$

<u>Four-point Cubic approximation</u>

Given ;(X_1, f_1),(X_2, f_2),(X_3, f_3) and (X_4, f_4)

For convenience, define the following terms:

$$Q_1 = X_3^3 \ (X_2 - X_1) - x_2^3 (X_3 - X_1) + X_1^3 (X_3 - X_2) \tag{B.10}$$

$$Q_2 = X_4^3 \ (X_2 - X_1) - x_2^3 (X_4 - X_1) + X_1^3 (X_4 - X_2) \tag{B.11}$$

$$Q_3 = (X_3 - X_2)(X_2 - X_1)(X_3 - X_1) \tag{B.12}$$

$$Q_4 = (X_4 - X_2)(X_2 - X_1)(X_4 - X_1) \tag{B.13}$$

$$Q_5 = f_3 \ (X_2 - X_1) - f_2 \ (X_3 - X_1) + f_1 \ (X_3 - X_2) \tag{B.14}$$

$$Q_5 = f_4 \ (X_2 - X_1) - f_2 \ (X_4 - X_1) + f_1 \ (X_4 - X_2) \tag{B.15}$$

In terms of Q_1 through Q_6 , the coefficients now become

$$a_3 = \frac{Q_3\ Q_6 - Q_4\ Q_5}{Q_2\ Q_3 - Q_1\ Q_4} \tag{B.16}$$

$$a_2 = \frac{Q_5 - a_3\ Q_1}{Q_3} \tag{B.17}$$

$$a_1 = \frac{f_2 - f_1}{X_2 - X_1} - a_3\ \frac{X_2^{\ 2} - X_1^{\ 3}}{X_2 - X_1} - a_2\ (X_1 + X2) \tag{B.18}$$

$$a_0 = f_1 - a_1\ X_1 - a_2\ X_1^{\ 2} - a_3\ X_1^{\ 3} \tag{B.19}$$

APPENDIX C
MAIN COMPUTRE PROGRAM

```
/*************************************************************/
/** for cube => rotate 1 deg : Normal force sdensors only**/
/*************************************************************/
#include <math.h>
#include <stdio.h>
#include "extra.h"
#define sq(x) x*x
#define tri(x) x*x*x
#define e 0.000000001

double tf1[6][6],tf2[6][6],tf3[6][6],ft1[6][6],ft2[6][6],
       ft3[6][6],v,mu,dx,dy,dz,r,w,pi,rad,f2x[3],f2y[3],
       f2z[3],m2z[3],bmax,bmin,f_ratio[3][6];
int *lpBasis,l;

void main()
{
   double f_unexp[6],t;
   int i,n,itr,j;
   double r1[3][3],r2[3][3],r3[3][3],ry[3][3],rz[3][3],
          rx[3][3],a1[3][3],a2[3][3],a3[3][3],p1[3],p2[3],
          p3[3],fmax,Nmax,check11,check12,check21,check22,
          check31,check32,x[3][3],m1[27][42],sol1[42],
          rhs1[27],cost1[42],obj1,f1x[3],f1y[3],f1z[3],
          m1z[3],fzmax,m2[29][44],sol2[44],rhs2[29],cost2[44],
          obj2,b11,b12,b21,b22,fio[6],b31,b32,
          ff[3][6],ffo[3][6],cn,cf,fi[6],ro[3][3],rcu[3][3],
          fmin,ratio,I3[3][3],del_f1,del_f2,del_f3,
          fzstart,txc,tyc,tzc,tx,ty,tz,fminx,fminy,fminz,
          txo,tyo,tzo,fexo[6],fexto[6],thx,thy,thz,
          ts1,tfi1,w1,fe1,f1,period1,ts2,tfi2,w2,fe2,f2,
          f_etra1[4],f_toto[6],f_etra2[4],f_etra3[4],
          period2,ts3,tfi3,w3,fe3,f3,ffmax,fosensed,period3,
          fie[6],f_tot[6],fex,f_ext[6],f_sensed,
          forg,del_t,count,f_ref,sensed;
   double extra();
   int nrow,ncol,error,itre,trakye,itpol;
   void golp(); void step1(); void step2(); void step3();

   FILE *fp1,*fp2,*fp3,*fopen();
/**
   fp1 = fopen("casrti5.d1","w");
```

```
   fp2 = fopen("casrti5.d2","w");
   fprintf(fp1,"expected Disturb.\n\n t fx fy fz mx my mz
                                                        \n\n");
   fprintf(fp2,"TOTAL Disturb.\n\n t fx fy fz mx my mz \n\n");
   fprintf(fp3,"Time thetax thetay thetaz forg
                                                 Nocor.ratio\n\n");
**/
   fp3 = fopen("Normal1.d3","w");
    /** Variables **/
/*
    pi[0]==> x position of i_th finger
    pi[1]==> y position of i_th finger
    pi[2]==> z position of i_th finger
    fix[j]===> force of x direction of j_th finger of i_th
               step.
    fiy[j]===> force of y direction of j_th finger of i_th
               step.
    fiz[j]===> force of z direction of j_th finger of i_th
               step.
    ri[][]  ==> rotation matrix of i_th finger from OBJ to
               contact center.
    tfi[][]==> Transformation matrix of i_th finger from OBJ
               to finger.
    fti[][]==> Transformation matrix of i_th finger from
               finger to OBJ.
    t ==>      Current Time from the start of the operation
    del_t ==>  Interval between the iterations == 0.01 sec
    f_exp[]  ==> Expected disturbance vector on the object
               system
    f_unexp[]==> Unexpected disturbance vector on the object
    f_tot[] ==> Total disturbance (f_exp + f_unexp)
    fz[i] ==> Sensed normal force from i_th fingertip
    mz[] ==>   Senced moment From the sensor system refered
               Obj.
    f_i[]==> Local wrench of the i_th finger
*/
/** Define pi the modulus of a circle **/
    pi = (double) acos(-1.0);
/** Size of the object  **/
    dx = 0.1; dy = 0.1; dz = 0.1;
    Nmax = 0.0;
/** Size of the Contact Area **/
    mu = 0.3;
    rad= sqrt(0.0001/ (double) acos(-1.0)); /**Radi. of the
                                                contact area**/
    v=3.0/(2.0*pi*(rad)*mu);  /** Propotionality const. **/
/** Pre planned Finger Position ==> Input Data**/
    x[0][0]=-dx/2.0;x[0][1]=0.0; x[0][2]=0.0;
    x[1][0]=0.00392950241; x[1][1]=-dy/2.0;
    x[1][2]=0.0142992626;
    x[2][0]=dx/2.0; x[2][1]=0.0; x[2][2]=0.0;
```

```
p1[0]=x[0][0];p1[1]=x[0][1];p1[2]=x[0][2];
p2[0]=x[1][0];p2[1]=x[1][1];p2[2]=x[1][2];
p3[0]=x[2][0];p3[1]=x[2][1];p3[2]=x[2][2];

for ( i = 0 ; i < 3 ; i ++ )
 for ( j = 0 ; j < 3 ; j ++ )
 {     I3[i][j] = 0.0 ;
       if( i == j )
            I3[i][j] = 1.0 ;
 }

/** Make the rotation matrices for each finger points **/

rotate(rx,-90.0,'x');
rotate(rz,-90.0,'z');
mult3(r1,rx,rz);
rotate(ry,90.0,'y');
rotate(rz,90.0,'z');
mult3(r2,ry,rz);
rotate(rx,-90.0,'x');
rotate(rz,90.0,'z');
mult3(r3,rx,rz);
mult(p1,r1,p1);
mult(p2,r2,p2);
mult(p3,r3,p3);
for(i=0;i<3;i++)
 {  p1[i]=-p1[i];
    p2[i]=-p2[i];
    p3[i]=-p3[i];
 }
a1[0][0]=p1[1]*r1[2][0]-p1[2]*r1[1][0];
a1[1][0]=p1[2]*r1[0][0]-p1[0]*r1[2][0];
a1[2][0]=p1[0]*r1[1][0]-p1[1]*r1[0][0];
a1[0][1]=p1[1]*r1[2][1]-p1[2]*r1[1][1];
a1[1][1]=p1[2]*r1[0][1]-p1[0]*r1[2][1];
a1[2][1]=p1[0]*r1[1][1]-p1[1]*r1[0][1];
a1[0][2]=p1[1]*r1[2][2]-p1[2]*r1[1][2];
a1[1][2]=p1[2]*r1[0][2]-p1[0]*r1[2][2];
a1[2][2]=p1[0]*r1[1][2]-p1[1]*r1[0][2];
a2[0][0]=p2[1]*r2[2][0]-p2[2]*r2[1][0];
a2[1][0]=p2[2]*r2[0][0]-p2[0]*r2[2][0];
a2[2][0]=p2[0]*r2[1][0]-p2[1]*r2[0][0];
a2[0][1]=p2[1]*r2[2][1]-p2[2]*r2[1][1];
a2[1][1]=p2[2]*r2[0][1]-p2[0]*r2[2][1];
a2[2][1]=p2[0]*r2[1][1]-p2[1]*r2[0][1];
a2[0][2]=p2[1]*r2[2][2]-p2[2]*r2[1][2];
a2[1][2]=p2[2]*r2[0][2]-p2[0]*r2[2][2];
a2[2][2]=p2[0]*r2[1][2]-p2[1]*r2[0][2];
a3[0][0]=p3[1]*r3[2][0]-p3[2]*r3[1][0];
a3[1][0]=p3[2]*r3[0][0]-p3[0]*r3[2][0];
a3[2][0]=p3[0]*r3[1][0]-p3[1]*r3[0][0];
a3[0][1]=p3[1]*r3[2][1]-p3[2]*r3[1][1];
```

```
a3[1][1]=p3[2]*r3[0][1]-p3[0]*r3[2][1];
a3[2][1]=p3[0]*r3[1][1]-p3[1]*r3[0][1];
a3[0][2]=p3[1]*r3[2][2]-p3[2]*r3[1][2];
a3[1][2]=p3[2]*r3[0][2]-p3[0]*r3[2][2];
a3[2][2]=p3[0]*r3[1][2]-p3[1]*r3[0][2];

for(i=0;i<6;i++)
 for(j=0;j<6;j++)
    { ft1[i][j]=0.0;
      ft2[i][j]=0.0;
      ft3[i][j]=0.0;
    }
for(i=0;i<3;i++)
 for(j=0;j<3;j++)
    {  ft1[i][j]=r1[i][j];
       ft1[i+3][j+3]=r1[i][j];
       ft1[i+3][j]=a1[i][j];
    }
for(i=0;i<3;i++)
 for(j=0;j<3;j++)
    {  ft2[i][j]=r2[i][j];
       ft2[i+3][j+3]=r2[i][j];
       ft2[i+3][j]=a2[i][j];
    }
for(i=0;i<3;i++)
 for(j=0;j<3;j++)
    {  ft3[i][j]=r3[i][j];
       ft3[i+3][j+3]=r3[i][j];
       ft3[i+3][j]=a3[i][j];
    }

for(i=0;i<6;i++)
 for(j=0;j<6;j++)
    { tf1[i][j]=0.0;
      tf2[i][j]=0.0;
      tf3[i][j]=0.0;
    }
for(i=0;i<3;i++)
 for(j=0;j<3;j++)
    { tf1[i][j]=r1[j][i];
      tf1[i+3][j+3]=r1[j][i];
      tf1[i+3][j]=a1[j][i];
    }
for(i=0;i<3;i++)
 for(j=0;j<3;j++)
    { tf2[i][j]=r2[j][i];
      tf2[i+3][j+3]=r2[j][i];
      tf2[i+3][j]=a2[j][i];
    }
for(i=0;i<3;i++)
 for(j=0;j<3;j++)
    { tf3[i][j]=r3[j][i];
```

```
        tf3[i+3][j+3]=r3[j][i];
        tf3[i+3][j]=a3[j][i];
      }
    /** Find the first grasp ratio **/
    step1(v,tf1,tf2,tf3,&rhs1,&cost1,&m1);
    nrow = 25; ncol = 42;
    golp( nrow, ncol, m1, cost1, rhs1,&sol1, &obj1, &error);
    if( ! error ){
     f1x[0] = sol1[0]-sol1[1];
     f1y[0] =  sol1[2]-sol1[3];
     f1z[0] =  sol1[4];
     m1z[0] =  sol1[5]-sol1[6];
     f1x[1] =  sol1[7]-sol1[8];
     f1y[1] =  sol1[9]-sol1[10];
     f1z[1] =  sol1[11];
     m1z[1] =  sol1[12]-sol1[13];
     f1x[2] =  sol1[14]-sol1[15];
     f1y[2] =  sol1[16]-sol1[17];
     f1z[2] =  sol1[18];
     m1z[2] =  sol1[19]-sol1[20];
/**  printf("step 1: fmax = %f Mmax = %f OBJ =
                              %f\n",sol1[21],sol1[22],obj1);
     printf("    f1x=%f    f1y=%f    f1z=%f    m1z=%f
                        \n",f1x[0],f1y[0],f1z[0],m1z[0]);
     printf("    f2x=%f    f2y=%f    f2z=%f    m2z=%f
                        \n",f1x[1],f1y[1],f1z[1],m1z[1]);
     printf("    f3x=%f    f3y=%f    f3z=%f    m3z=%f
                        \n",f1x[2],f1y[2],f1z[2],m1z[2]);
**/
    }
    else {
     Nmax = 100000000;
     goto step6;
    }

/*****  Check the frictional constraints for each finger  ***/
    check11 = (double)sqrt(f1x[0]*f1x[0]+f1y[0]*f1y[0]);
    check12 = v*dabs(m1z[0]);
    check21 = (double)sqrt(f1x[1]*f1x[1]+f1y[1]*f1y[1]);
    check22 = v*dabs(m1z[1]);
    check31 = (double)sqrt(f1x[2]*f1x[2]+f1x[2]*f1x[2]);
    check32 = v*dabs(m1z[2]);
    if( (mu*f1z[0]) < check11 || (mu*f1z[0]) < check12 )
     {
       for(i=0;i<3;i++)
       {  f1x[i]=0.0;
          f1y[i]=0.0;
          f1z[i]=0.0;
          m1z[i]=0.0;   }
       goto step2;
     }
    if( (mu*f1z[1]) < check21 || (mu*f1z[1]) < check22 )
```

```
      {
        for(i=0;i<3;i++)
        {  f1x[i]=0.0;
           f1y[i]=0.0;
           f1z[i]=0.0;
           m1z[i]=0.0;   }
        goto step2;
      }
     if( (mu*f1z[2]) < check31 || (mu*f1z[2]) < check32 )
      {
        for(i=0;i<3;i++)
        {  f1x[i]=0.0;
           f1y[i]=0.0;
           f1z[i]=0.0;
           m1z[i]=0.0;   }
        goto step2;
      }

   /**** Calculate the First Grasping Ratio ***/

     fzmax = f1z[0];
     if(f1z[1] > fzmax ) fzmax = f1z[1];
     if(f1z[2] > fzmax ) fzmax = f1z[2];
     for(i=0;i<3;i++)
     {   f1x[i] = f1x[i]/fzmax;
         f1y[i] = f1y[i]/fzmax;
         f1z[i] = f1z[i]/fzmax;
         m1z[i] = m1z[i]/fzmax;
     }
/**
     printf(" First ratio = %f \n",fzmax);
      printf("    f1x=%f    f1y=%f    f1z=%f    m1z=%f
                   \n",f1x[0],f1y[0],f1z[0],m1z[0]);
      printf("    f2x=%f    f2y=%f    f2z=%f    m2z=%f
                   \n",f1x[1],f1y[1],f1z[1],m1z[1]);
      printf("    f3x=%f    f3y=%f    f3z=%f    m3z=%f
                   \n",f1x[2],f1y[2],f1z[2],m1z[2]);
**/
/**** If any finger has no normal force, another ratio can be
calculated for which all fingers are constrainted to be
active.****/

     if( f1z[0] == 0.0 ) goto step2;
     else if( f1z[1] == 0.0 ) goto step2;
     else if( f1z[2] == 0.0 ) goto step2;
     else
     {    for(i=0;i<3;i++)
       {  f2x[i]=f1x[i];
          f2y[i]=f1y[i];
          f2z[i]=f1z[i];
          m2z[i]=m1z[i];
       }
```

```
      goto step23;
    }
step2:
    step2(v,tf1,tf2,tf3,&rhs2,&cost2,&m2);
    nrow = 27; ncol = 44;
    golp( nrow, ncol, m2, cost2, rhs2,&sol2, &obj2, &error);
    if( ! error )
    {   f2x[0] =  sol2[0]-sol2[1];
        f2y[0] =  sol2[2]-sol2[3];
        f2z[0] =  sol2[4];
        m2z[0] =  sol2[5]-sol2[6];
        f2x[1] =  sol2[7]-sol2[8];
        f2y[1] =  sol2[9]-sol2[10];
        f2z[1] =  sol2[11];
        m2z[1] =  sol2[12]-sol2[13];
        f2x[2] =  sol2[14]-sol2[15];
        f2y[2] =  sol2[16]-sol2[17];
        f2z[2] =  sol2[18];
        m2z[2] =  sol2[19]-sol2[20];
    }
    else
    {   Nmax = 100000000;
        goto step6;
    }
step23:
    bmax = f2z[0]; bmin = 1.0;
    for(i = 1; i < 3; i++ )
          if( bmax < f2z[i] ) bmax = f2z[i];

/*** Adjust the normal forces so that fingers can provide the
required frictional forces at the moment ***/

    check11 = (double)sqrt(sq(f2x[0])+sq(f2y[0]));
    check12 = v*dabs(sol2[5]-sol2[6]);
    check21 = (double)sqrt(sq(f2x[1])+sq(f2y[1]));
    check22 = v*dabs(sol2[12]-sol2[13]);
    check31 = (double)sqrt(sq(f2x[2])+sq(f2y[2]));
    check32 = v*dabs(sol2[19]-sol2[20]);
    b11 = check11/(mu*f2z[0]);
    b21 = check21/(mu*f2z[1]);
    b31 = check31/(mu*f2z[2]);
    b12 = check12/(mu*f2z[0]);
    b22 = check22/(mu*f2z[1]);
    b32 = check32/(mu*f2z[2]);
    bmax = b11;
    if( bmax < b12 ) bmax = b12;
    if( bmax < b21 ) bmax = b21;
    if( bmax < b22 ) bmax = b12;
    if( bmax < b31 ) bmax = b31;
    if( bmax < b32 ) bmax = b32;
  /**** Calculate the Second Grasping Ratio ***/
    for(i=0;i<3;i++)
```

```
    {   f2x[i] = f1x[i]+(f2x[i]-f1x[i])/bmax;
        f2y[i] = f1y[i]+(f2y[i]-f1y[i])/bmax;
        f2z[i] = f1z[i]+(f2z[i]-f1z[i])/bmax;
        m2z[i] = m1z[i]+(m2z[i]-m1z[i])/bmax;
    }
    for(i = 0; i < 3 ; i++ )
        if(f2z[i] < bmin)  { bmin = f2z[i]; l=i; }
/**
    printf(" Second ratio = %f \n",bmin);
     printf("    f1x=%f    f1y=%f    f1z=%f    m1z=%f
                    \n",f2x[0],f2y[0],f2z[0],m2z[0]);
     printf("    f2x=%f    f2y=%f    f2z=%f    m2z=%f
                    \n",f2x[1],f2y[1],f2z[1],m2z[1]);
     printf("    f3x=%f    f3y=%f    f3z=%f    m3z=%f
                    \n",f2x[2],f2y[2],f2z[2],m2z[2]);
**/
    check11 = (double)sqrt(f2x[0]*f2x[0]+f2y[0]*f2y[0]);
    check12 = v*dabs(m2z[0]);
    check21 = (double)sqrt(f2x[1]*f2x[1]+f2y[1]*f2y[1]);
    check22 = v*dabs(m2z[1]);
    check31 = (double)sqrt(f2x[2]*f2x[2]+f2y[2]*f2y[2]);
    check32 = v*dabs(m2z[2]);
    if( mu*f2z[0] < check11 && dabs(mu*f2z[0] - check11) > e)
    {      Nmax = 100000000;
           goto step6;
    }
    if ( mu*f2z[0] < check12 && dabs(mu*f2z[0]-check12) > e )
    {      Nmax = 100000000;
           goto step6;
    }
    if( mu*f2z[1] < check21 && dabs(mu*f2z[1] - check21) > e)
    {      Nmax = 100000000;
           goto step6;
    }
    if ( mu*f2z[1] < check22 && dabs(mu*f2z[1]-check22) > e )
    {      Nmax = 100000000;
           goto step6;
    }
    if( mu*f2z[2] < check31 && dabs(mu*f2z[2] - check31) > e)
    {      Nmax = 100000000;
           goto step6;
    }
    if ( mu*f2z[2] < check32 && dabs(mu*f2z[2]-check32) > e )
    {      Nmax = 100000000;
           goto step6;
    }

/** Initial orientation of the object **/

    for ( i = 0 ; i < 3 ; i ++ )
     for ( j = 0 ; j < 3 ; j ++ )
           rcu[i][j] = I3[i][j];
```

```
/** Input Given Expected Disturbance at this moment **/

    ratio = 0.0;
    t = 0.0;
    itr = 0;
    ts1 = 0.21; /** start the first disturbance **/
    ts2 = 0.51; /** start the second disturbance **/
    ts3 = 1.21; /** start the third disturbance **/
    tfi1 = 0.31;/** start to reduce the first disturb.**/
    tfi2 = 0.71;/** start to reduce the second disturb.**/
    tfi3 = 1.41;/** start to reduce the third disturb.**/
    f1 = 10.0;/** Max. of the first disturb.**/
    f2 = -0.35;/** Max. of the ssecond disturb.**/
    f3 = -0.2; /** Max. of the third disturb. **/
    period1 = 0.4 ;
    period2 = 0.4 ;
    period3 = 0.4 ;
    w1 = 2.0*pi/period1;
    w2 = 2.0*pi/period2;
    w3 = 2.0*pi/period3;
    del_t = 0.01;
    forg = ffmax=0.0;
    thx = thy = thz = 0.0;
    tx=ty=tz=txc=tyc=tzc=0.0;
    while ( t < 1.7 )
    {
       t = itr*del_t;
       printf("\n\nTime T = %f\n",t);
       for( j = 0 ; j < 6 ; j++ )
         fi[j] = 0.0 ;
       printf(    "fx=%f    fy=%f    fz=%f    mx=%f    my=%f
               mz=%f\n",fi[0],fi[1],fi[2],fi[3],fi[4],fi[5]);
/**
       fprintf(fp1,"%f %f %f %f %f %f %f\n",t,fi[0],fi[1],
                                   fi[2],fi[3],fi[4],fi[5]);
**/
       for ( i = 0 ; i < 6 ; i ++ )
         fie[i] = 0.0 ;
       fe1 = fe2 = fe3 = fex = 0.0;
       if ( t > ts1 && t <= (ts1+period1/4.0 ))
       {   fe1 = f1*sin(w1*(t-ts1));
           trakye = 1;
       }
       else if ( t > (ts1+period1/4.0 ) && t <= tfi1 )
         fe1 = f1;
       else if ( t > tfi1 && t < tfi1+period1/4.0 )
         fe1 = f1*cos(w1*(t-tfi1));
       else if ( t >= (tfi1+period1/4.0) && t <= ts2 )
       {    fe1 = 0.0;
            trakye = 0;
       }
       if ( t > ts2 && t <= (ts2+period2/4.0 ))
```

```
        {    fe2 = f2*sin(w2*(t-ts2));
             trakye = 2;
        }
        else if ( t > (ts2+period2/4.0 ) && t <= tfi2 )
          fe2 = f2;
        else if ( t > tfi2 && t < tfi2+period2/4.0 )
          fe2 = f2*cos(w2*(t-tfi2));
        else if ( t >= (tfi2+period2/4.0) && t < ts3 )
        {     fe2 = 0.0;
              trakye = 0;
        }
        if ( t > ts3 && t <= (ts3+period3/4.0 ))
        {    fe3 = f3*sin(w3*(t-ts3));
             trakye = 3;
        }
        else if ( t > (ts3+period3/4.0 ) && t <= tfi3 )
              fe3 = f3;
        else if ( t > tfi3 && t < tfi3+period3/4.0 )
              fe3 = f3*cos(w3*(t-tfi3));
        else if ( t >= (tfi3+period3/4.0) )
        {     fe3 = 0.0;
              trakye = 0;
        }
        fex = fe1+fe2+fe3;

/** Sumation  all of the external disturbance **/

        fie[0] = 0.0;
        fie[1] = 0.0;
        if ( trakye == 1 )
             fie[2] = fex;
        if ( trakye == 2 )
             fie[4] = fex;
        if ( trakye == 3 )
             fie[5] = fex;
        fie[3] = 0.0;
              printf("    %f    %f    %f    %f    %f
                %f\n",fie[0],fie[1],fie[2],fie[3],
                                         fie[4],fie[5]);

        for ( i = 0 ; i < 6 ; i ++ )
              f_toto[i] = fi[i] + fie[i] ;
        printf("t  =  %f  Tot  fx=%f  fy=%f  fz=%f  mx=%f  my=%f
                mz=%f\n",t,f_toto[0],f_toto[1],f_toto[2],
                              f_toto[3],f_toto[4],f_toto[5]);
/**
         fprintf(fp2,"%f        %f    %f    %f    %f    %f
                     %f\n",t,f_toto[0],f_toto[1],
                     f_toto[2],f_toto[3],f_toto[4],f_toto[5]);
**/
        for ( i = 0 ; i < 3 ; i ++ )
                printf("rcu   =   %f        %f
```

```
                    %f\n",rcu[i][0],rcu[i][1],rcu[i][2]);

f_sensed = 0.0;
fosensed = 0.0;
sensed = 0.0;
itr += 1;
for ( i = 0 ; i < 6 ; i ++ )
{   f_tot[i] = fi[i] + fie[i] ;
    sensed += dabs(f_toto[i]);
    f_sensed += dabs(f_tot[i]);
    fosensed += dabs(fi[i]);
}
if ( f_sensed == 0.0  && sensed == 0.0 )
{   for ( i = 0 ; i < 3 ; i ++ )
     for ( j = 0 ; j < 3 ; j ++ )
         rcu[i][j] = I3[i][j];
         tx = ty = tz = 0.0;
         thx = thy = thz = 0.0;
         ffmax = forg = 0.0;
         for( i = 0 ; i < 3 ; i ++ )
               for ( j = 0 ; j < 3 ; j ++ )
                     ffo[i][j] = 0.0;
         for ( i = 0 ; i < 6 ; i ++ )
               fexo[i] = f_toto[i];
         goto cont;
}
if ( f_sensed == 0.0   )
{   for ( i = 0 ; i < 3 ; i ++ )
        for ( j = 0 ; j < 3 ; j ++ )
              rcu[i][j] = I3[i][j];
    tx = ty = tz = 0.0;
    thx = thy = thz = 0.0;
    ffmax = forg = 0.0;
    for ( i = 0 ; i < 6 ; i ++ )
        fexo[i] = f_toto[i];
    goto cont ;
}
if ( fex == 0.0 )
{   for ( i = 0 ; i < 3 ; i ++ )
        for ( j = 0 ; j < 3 ; j ++ )
            rcu[i][j] = I3[i][j];
    if((dabs(f_toto[0])+dabs(f_toto[1])+
                        dabs(f_toto[2])) == 0.0 )
        f_toto[1] = - 0.000000001;
    check(f_toto,ff,&ffmax);
    if ( ratio < ffmax )
         ratio = ffmax ;
    forg = ffmax ;
    tx = ty = tz = 0.0;
    thx = thy = thz = 0.0;
    printf("time=%f tx= %f ty= %f tz= %f required=%f No
                        forg.=%f \n",t,tx,ty,tz,ffmax,forg);
```

```
            for ( i = 0 ; i < 6 ; i ++ )
                 fexo[i] = f_toto[i];
            goto cont;
        }
        else
        {
         if((dabs(fexo[0])+dabs(fexo[1])+dabs(fexo[2]))==0.0 )
             fexo[1] = - 0.000000001;
         f_ext[0] = rcu[0][0]*fexo[0]+
                          rcu[0][1]*fexo[1]+rcu[0][2]*fexo[2];
         f_ext[1] = rcu[1][0]*fexo[0]+
                          rcu[1][1]*fexo[1]+rcu[1][2]*fexo[2];
         f_ext[2] = rcu[2][0]*fexo[0]+
                          rcu[2][1]*fexo[1]+rcu[2][2]*fexo[2];
         f_ext[3] = rcu[0][0]*fexo[3]+
                          rcu[0][1]*fexo[4]+rcu[0][2]*fexo[5];
         f_ext[4] = rcu[1][0]*fexo[3]+
                          rcu[1][1]*fexo[4]+rcu[1][2]*fexo[5];
         f_ext[5] = rcu[2][0]*fexo[3]+
                          rcu[2][1]*fexo[4]+rcu[2][2]*fexo[5];
         check(f_ext,ffo,&fmax);
        }
        if((dabs(f_toto[0])+dabs(f_toto[1])+
                                      dabs(f_toto[2])) == 0.0 )
           f_toto[1] = - 0.000000001;
        f_ext[0] = rcu[0][0]*f_toto[0]+rcu[0][1]*f_toto[1]+
                                           rcu[0][2]*f_toto[2];
        f_ext[1] = rcu[1][0]*f_toto[0]+rcu[1][1]*f_toto[1]+
                                           rcu[1][2]*f_toto[2];
        f_ext[2] = rcu[2][0]*f_toto[0]+rcu[2][1]*f_toto[1]+
                                           rcu[2][2]*f_toto[2];
        f_ext[3] = rcu[0][0]*f_toto[3]+rcu[0][1]*f_toto[4]+
                                           rcu[0][2]*f_toto[5];
        f_ext[4] = rcu[1][0]*f_toto[3]+rcu[1][1]*f_toto[4]+
                                           rcu[1][2]*f_toto[5];
        f_ext[5] = rcu[2][0]*f_toto[3]+rcu[2][1]*f_toto[4]+
                                           rcu[2][2]*f_toto[5];
        check(f_ext,ff,&ffmax);
        if ( ffmax > ratio )
           ratio = ffmax ;
        check(f_toto,ff,&forg);
         printf("time=%f    required=%f    No    cor.=%f
                              ra=%f\n",t,ffmax,forg,ratio);

/*Monitor the Normal force changes for the next step */

next:
        if((dabs(f_tot[0])+dabs(f_tot[1])+dabs(f_tot[2]))==0.0)
             f_tot[1] = - 0.000000001;
        f_ext[0] = rcu[0][0]*f_tot[0]+rcu[0][1]*f_tot[1]+
                                           rcu[0][2]*f_tot[2];
        f_ext[1] = rcu[1][0]*f_tot[0]+rcu[1][1]*f_tot[1]+
```

```
                                    rcu[1][2]*f_tot[2];
     f_ext[2] = rcu[2][0]*f_tot[0]+rcu[2][1]*f_tot[1]+
                                    rcu[2][2]*f_tot[2];
     f_ext[3] = rcu[0][0]*f_tot[3]+rcu[0][1]*f_tot[4]+
                                    rcu[0][2]*f_tot[5];
     f_ext[4] = rcu[1][0]*f_tot[3]+rcu[1][1]*f_tot[4]+
                                    rcu[1][2]*f_tot[5];
     f_ext[5] = rcu[2][0]*f_tot[3]+rcu[2][1]*f_tot[4]+
                                    rcu[2][2]*f_tot[5];
     check(f_ext,ff,&fmax);

/** Calculate the change of the normal forces ***/

     del_f1 = ff[0][2] - ffo[0][2];
     del_f2 = ff[1][2] - ffo[1][2];
     del_f3 = ff[2][2] - ffo[2][2];

     for ( i = 0 ; i < 6 ; i ++ )
      for ( j = 0 ; j < 3 ; j ++ )
          ro[i][j] = rcu[j][i];
     mult3(ro,ro,rcu);
     fexo[0] = ro[0][0]*f_ext[0]+
                     ro[0][1]*f_ext[1]+ro[0][2]*f_ext[2];
     fexo[1] = ro[1][0]*f_ext[0]+
                     ro[1][1]*f_ext[1]+ro[1][2]*f_ext[2];
     fexo[2] = ro[2][0]*f_ext[0]+
                     ro[0][1]*f_ext[1]+ro[2][2]*f_ext[2];
     fexo[3] = ro[0][0]*f_ext[3]+
                     ro[0][1]*f_ext[4]+ro[0][2]*f_ext[5];
     fexo[4] = ro[1][0]*f_ext[3]+
                     ro[0][1]*f_ext[4]+ro[1][2]*f_ext[5];
     fexo[5] = ro[2][0]*f_ext[3]+
                     ro[0][1]*f_ext[4]+ro[2][2]*f_ext[5];

/** Add normal force changes as a next disturbance for next**/

     printf("df1  =  %f  df2  =  %f  df3  =  %f  \n",
                                    del_f1,del_f2,del_f3);
     if(dabs(del_f1)<e && dabs(del_f2)<e && dabs(del_f3)<e)
     {   printf("KEEP the POSITION \n");
         tx=ty=tz=0.0;
         txc = tyc = tzc = 0.0;
         goto cont;
     }
     fie[0] = tf1[0][2]*del_f1+tf2[0][2]*del_f2+
                                         tf3[0][2]*del_f3;
     fie[1] = tf1[1][2]*del_f1+tf2[1][2]*del_f2+
                                         tf3[1][2]*del_f3;
     fie[2] = tf1[2][2]*del_f1+tf2[2][2]*del_f2+
                                         tf3[2][2]*del_f3;
     fie[3] = tf1[3][2]*del_f1+tf2[3][2]*del_f2+
                                         tf3[3][2]*del_f3;
```

```
        fie[4] = tf1[4][2]*del_f1+tf2[4][2]*del_f2+
                                              tf3[4][2]*del_f3;
        fie[5] = tf1[5][2]*del_f1+tf2[5][2]*del_f2+
                                              tf3[5][2]*del_f3;

        printf(" %f %f %f %f %f %f\n",fie[0],fie[1],
                               fie[2],fie[3],fie[4],fie[5]);
        for ( i = 0 ; i < 6 ; i ++ )
           f_tot[i] = f_toto[i] + fie[i] ;

        printf("Time = %f\n Estimated = %f %f %f %f %f %f\n",
                t+del_t,f_tot[0],f_tot[1],f_tot[2],
                               f_tot[3],f_tot[4],f_tot[5]);

/** Start Trial move to prepare for the next time **/

        f_ext[0] = rcu[0][0]*f_tot[0]+rcu[0][1]*f_tot[1]+
                                         rcu[0][2]*f_tot[2];
        f_ext[1] = rcu[1][0]*f_tot[0]+rcu[1][1]*f_tot[1]+
                                         rcu[1][2]*f_tot[2];
        f_ext[2] = rcu[2][0]*f_tot[0]+rcu[2][1]*f_tot[1]+
                                         rcu[2][2]*f_tot[2];
        f_ext[3] = rcu[0][0]*f_tot[3]+rcu[0][1]*f_tot[4]+
                                         rcu[0][2]*f_tot[5];
        f_ext[4] = rcu[1][0]*f_tot[3]+rcu[1][1]*f_tot[4]+
                                         rcu[1][2]*f_tot[5];
        f_ext[5] = rcu[2][0]*f_tot[3]+rcu[2][1]*f_tot[4]+
                                         rcu[2][2]*f_tot[5];

        printf( "fx=%f fy=%f fz=%f mx=%f my=%f mz=%f\n\n",
           f_ext[0],f_ext[1],
                          f_ext[2],f_ext[3],f_ext[4],f_ext[5]);

        check(f_ext,ff,&fmax);
        fzstart = fmax ;
        fmin = fmax ;
        fzmax = fmax;
        tx = ty = tz = 1.0;
        txc = tyc = tzc = 0.0;


/** Search For the Best Direction to Rotate using sensed
                               normal force  ONLY  **/
        rotate ( rx, 1.0 , 'x' );
        mult3( ro,rx,rcu);
        f_ext[0] = ro[0][0]*f_tot[0]+
                          ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
        f_ext[1] = ro[1][0]*f_tot[0]+
                          ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
        f_ext[2] = ro[2][0]*f_tot[0]+
                          ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
        f_ext[3] = ro[0][0]*f_tot[3]+
```

```
                      ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
f_ext[4] = ro[1][0]*f_tot[3]+
                      ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
f_ext[5] = ro[2][0]*f_tot[3]+
                      ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
check(f_ext,ff,&fmax);
if ( fzstart > fmax )
{  fmin = fmax;
   fminx = fmax;
   txc = 1.0;
}
rotate ( r1, -1.0 , 'x' );
mult3( ro,r1,rcu);
f_ext[0] = ro[0][0]*f_tot[0]+
                      ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
f_ext[1] = ro[1][0]*f_tot[0]+
                      ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
f_ext[2] = ro[2][0]*f_tot[0]+
                      ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
f_ext[3] = ro[0][0]*f_tot[3]+
                      ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
f_ext[4] = ro[1][0]*f_tot[3]+
                      ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
f_ext[5] = ro[2][0]*f_tot[3]+
                      ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
check(f_ext,ff,&fmax);
if ( txc == 1.0 && fmax < fminx)
{    txc = -1.0;
     fmin = fmax;
}
else if ( txc == 0.0 && fmax < fzstart )
{    txc = -1.0;
     fmin = fmax ;
}
rotate ( ry, 1.0 , 'y' );
mult3( ro,ry,rcu);
f_ext[0] = ro[0][0]*f_tot[0]+
                      ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
f_ext[1] = ro[1][0]*f_tot[0]+
                      ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
f_ext[2] = ro[2][0]*f_tot[0]+
                      ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
f_ext[3] = ro[0][0]*f_tot[3]+
                      ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
f_ext[4] = ro[1][0]*f_tot[3]+
                      ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
f_ext[5] = ro[2][0]*f_tot[3]+
                      ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
check(f_ext,ff,&fmax);
if ( fzstart > fmax )
{  fminy = fmax;
   tyc = 1.0;
```

```
}
if ( fmin > fmax )
   fmin = fmax;
rotate ( r2, -1.0 , 'y' );
mult3( ro,r2,rcu);
f_ext[0] = ro[0][0]*f_tot[0]+
                    ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
f_ext[1] = ro[1][0]*f_tot[0]+
                    ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
f_ext[2] = ro[2][0]*f_tot[0]+
                    ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
f_ext[3] = ro[0][0]*f_tot[3]+
                    ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
f_ext[4] = ro[1][0]*f_tot[3]+
                    ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
f_ext[5] = ro[2][0]*f_tot[3]+
                    ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
check(f_ext,ff,&fmax);
if ( tyc == 1.0 && fmax < fminy)
   tyc = -1.0;
else if ( tyc == 0.0 && fmax < fzstart )
   tyc = -1.0;
if ( fmin > fmax )
   fmin = fmax;
rotate ( rz, 1.0 , 'z' );
mult3( ro,rz,rcu);
f_ext[0] = ro[0][0]*f_tot[0]+
                    ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
f_ext[1] = ro[1][0]*f_tot[0]+
                    ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
f_ext[2] = ro[2][0]*f_tot[0]+
                    ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
f_ext[3] = ro[0][0]*f_tot[3]+
                    ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
f_ext[4] = ro[1][0]*f_tot[3]+
                    ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
f_ext[5] = ro[2][0]*f_tot[3]+
                    ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
check(f_ext,ff,&fmax);
if ( fzstart > fmax )
 { fminz = fmax;
   tzc = 1.0;
 }
if ( fmin > fmax )
   fmin = fmax;
rotate ( r1, -1.0 , 'z' );
mult3( ro,r1,rcu);
f_ext[0] = ro[0][0]*f_tot[0]+
                    ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
f_ext[1] = ro[1][0]*f_tot[0]+
                    ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
f_ext[2] = ro[2][0]*f_tot[0]+
```

```
                         ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
      f_ext[3] = ro[0][0]*f_tot[3]+
                         ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
      f_ext[4] = ro[1][0]*f_tot[3]+
                         ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
      f_ext[5] = ro[2][0]*f_tot[3]+
                         ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
      check(f_ext,ff,&fmax);
      if ( tzc == 1.0 && fmax < fminz)
         tzc = -1.0;
      else if ( tzc == 0.0 && fmax < fzstart )
         tzc = -1.0;
      if ( fmin > fmax )
         fmin = fmax;
      printf("txc = %f tyc = %f tzc = %f\n",txc,tyc,tzc);
      txo = txc * tx /180.0 * pi;
      tyo = tyc * ty /180.0 * pi;
      tzo = tzc * tz /180.0 * pi;
      printf("tx = %f ty = %f tz = %f\n",txo,tyo,tzo);

/**  Choose angles with the axes which is improving grasp **/

      if ( fzmax == fmin )
      {    printf ("Don't Move\n");
           tx = ty = tz = 0.0;
           txc = tyc = tzc = 0.0;
           goto cont;
      }
      else
      {
          tx = txo;
          ty = tyo;
          tz = tzo;
          count = 1.0;
keepgo:

          printf("fzstart = %f \n",fzstart );
          printf("tx    =    %f    ty    =    %f    tz    =
                         %f\n",tx/pi*180,ty/pi*180,tz/pi*180);

          ro[0][0] = cos(tz)*cos(ty);
          ro[0][1] = cos(tz)*sin(ty)*sin(tx)-sin(tz)*cos(tx);
          ro[0][2] = sin(ty)*cos(tz)*cos(tx)+sin(tz)*sin(tx);
          ro[1][0] = sin(tz)*cos(ty);
          ro[1][1] = cos(tx)*cos(tz)+sin(tx)*sin(ty)*sin(tz);
          ro[1][2] = sin(tz)*sin(ty)*cos(tx)-sin(tx)*cos(tz);
          ro[2][0] = -sin(ty);
          ro[2][1] = cos(ty)*sin(tx);
          ro[2][2] = cos(tx)*cos(ty);
          mult3( ro,ro,rcu );
          f_ext[0] = ro[0][0]*f_tot[0]+
                         ro[0][1]*f_tot[1]+ro[0][2]*f_tot[2];
```

```
            f_ext[1] = ro[1][0]*f_tot[0]+
                            ro[1][1]*f_tot[1]+ro[1][2]*f_tot[2];
            f_ext[2] = ro[2][0]*f_tot[0]+
                            ro[2][1]*f_tot[1]+ro[2][2]*f_tot[2];
            f_ext[3] = ro[0][0]*f_tot[3]+
                            ro[0][1]*f_tot[4]+ro[0][2]*f_tot[5];
            f_ext[4] = ro[1][0]*f_tot[3]+
                            ro[1][1]*f_tot[4]+ro[1][2]*f_tot[5];
            f_ext[5] = ro[2][0]*f_tot[3]+
                            ro[2][1]*f_tot[4]+ro[2][2]*f_tot[5];
            check(f_ext,ff,&fmax);
            if ( fmax < fzstart )
            {     fzstart = fmax ;
                  txc = tx ;
                  tyc = ty ;
                  tzc = tz ;
                  count +=1.0;
                  tx = txo * count;
                  ty = tyo * count;
                  tz = tzo * count;
                  goto keepgo;
            }
            if( fmax > fzstart && count > 1.0 )
            {     ro[0][0] = cos(tzc)*cos(tyc);
                  ro[0][1] = cos(tzc)*sin(tyc)*sin(txc)-
                                        sin(tzc)*cos(txc);
                  ro[0][2] = sin(tyc)*cos(tzc)*cos(txc)+
                                        sin(tzc)*sin(txc);
                  ro[1][0] = sin(tzc)*cos(tyc);
                  ro[1][1] = cos(txc)*cos(tzc)+
                                   sin(txc)*sin(tyc)*sin(tzc);
                  ro[1][2] = sin(tzc)*sin(tyc)*cos(txc)-
                                        sin(txc)*cos(tzc);
                  ro[2][0] = -sin(tyc);
                  ro[2][1] = cos(tyc)*sin(txc);
                  ro[2][2] = cos(txc)*cos(tyc);
            }
            if (fmax > fzstart && count == 1 )
            {     for ( i = 0 ; i < 3 ; i ++ )
                        for ( j = 0 ; j < 3 ; j ++ )
                              ro[i][j] = I3[i][j];
                  tx = ty = tz = 0.0;
                  txc = tyc = tzc = 0.0;
            }
            mult3 (rcu,ro,rcu);
            tx = txc *180.0/pi;
            ty = tyc *180.0/pi;
            tz = tzc *180.0/pi;
cont:
            if((dabs(f_tot[0])+dabs(f_tot[1])+
                                        dabs(f_tot[2])) ==0.0)
                  f_tot[1] = - 0.000000001;
```

```
            f_ext[0] = rcu[0][0]*f_tot[0]+
                       rcu[0][1]*f_tot[1]+rcu[0][2]*f_tot[2];
            f_ext[1] = rcu[1][0]*f_tot[0]+
                       rcu[1][1]*f_tot[1]+rcu[1][2]*f_tot[2];
            f_ext[2] = rcu[2][0]*f_tot[0]+
                       rcu[2][1]*f_tot[1]+rcu[2][2]*f_tot[2];
            f_ext[3] = rcu[0][0]*f_tot[3]+
                       rcu[0][1]*f_tot[4]+rcu[0][2]*f_tot[5];
            f_ext[4] = rcu[1][0]*f_tot[3]+
                       rcu[1][1]*f_tot[4]+rcu[1][2]*f_tot[5];
            f_ext[5] = rcu[2][0]*f_tot[3]+
                       rcu[2][1]*f_tot[4]+rcu[2][2]*f_tot[5];
            printf("degree ==> x=%f y=%f z=%f\n",tx,ty,tz);
            check(f_ext,ff,&fmax);
            if ( f_sensed == 0 )
            forg = fmax;
            thx += tx;
            thy += ty;
            thz += tz;
            printf("Time   =   %f   force   for   the   next
                                 time=%f\n",t,fmax);
      }
        fprintf(fp3,"%f    %f    %f    %f    %f    %f
                           \n",t,thx,thy,thz,ffmax,forg);
   }
step6:
   printf("Nmax = %f\n",Nmax);
  }
```

REFERENCES

Akella, Prasad and Mark Cutkosky, 1989, "Manipulating with Soft Fingers 1: Modeling Contacts and Dynamics", IEEE CH2750-8/89/0000/0764, pp. 764-769.

Arimoto, S., F. Miyazaki and S. Kawamura, 1987, "Cooperative Motion Control of Multi Robot Arms or Fingers", IEEE CH2413-3/87/0000/1407, pp. 1407-1412.

Badreldin, Ashraf and Ali A. Seireg, 1988, "Optimizing the Handling of Objects with Multi-finger Grippers", Computers in Mechanical Engineering, Vol. 6, No. 5, pp. 47-57.

Badreldin, Ashraf and Ali A. Seireg, 1990, "Generalized Solution for the Optimal Prehension State Using End Effectors With Multi-Articulated Fingers", ASME Journal of Mechanical Design, Vol. 112, pp. 84-89.

Bejczy, Antal K., 1977, "Effect of Hand-Based Sensors on Manipulator Control Performance", Mechanism and Machine Theory, Vol. 12, pp. 547-567.

Brock, David L., 1988, "Enhancing the Dexterity of a Robot Hand using Controlled Slip",IEEE CH2555-1/88/0000/0249, pp. 249-251.

Brost, Randy C., 1988, " Automatic Grasp Planning in the Presence of Uncertainty", The Internal Journal of Robotics Research , Vol. 7, No. 1, pp. 3-17.

Cole, Arlene, John Hauser, and Shankar Sastry, 1988, "Kinematics and Control of Multifingered Hands with Rolling Contact", IEEE CH2555-1/88/0000/0228, pp. 228-233.

Cole, Arlene, Ping Hsu, and Shankar Sastry, 1989, "Dynamic Regrasping by Coordinated Control of Sliding for a Multifingered Hands", IEEE CH2750-8/89/0000/0781, 781-786.

Craig, J. J., 1986, Introduction to Robots, Mechanics and Control, Addison-Wesley Publishing Co., Reading, MA.

Crossley, F. R. Erskine, 1975, "Design for a Three-fingered Hand", Mechanism and Machine Theory, Vol. 12, pp. 85-93.

Cutkosky, M. R., 1985, Robotic Grasping and Fine Manipulation, Kluwer Academic Publishers, Boston, MA.

Cutkosky, M. R. and Paul K. Wright, 1986, "Friction, Stability and the Design of Robotic Fingers", The Internal Journal of Robotics Research, Vol. 5, No. 4, pp. 20-92.

Dario, Paolo and Giorigo Buttazzo, 1987, "An Anthropomorphic Robot Finger for Investigating Artificial Tactile Perception", The International Journal of Robotics Research, Vol. 6, No. 3, pp. 25-48.

Demmel, J. and G. Lafferriere, 1989, "Optimal Three Finger Grasps", IEEE CH2750-8/89/0000/0936, pp. 936-942.

El-Deen, Yehia Hosem 1986, "Computer Control of Vehicle Skidding On Slippery Roads", Ph.D. Dissertation, University of Wisconsin.

El-Deen, Yehia Hossam and Ali A. Seireg, 1987, "Mechatronics for Cars: Inter\grating Machines and Electronics to Prevent Skidding on Icy Roads", Computers in Mechanical Engineering , Vol. 5, No. 4, pp. 10-22.

Fearing, Ronald S., 1986, "Simplified Grasping and Manipulation with Dexterous Robot Hands", IEEE Journal of Robotics and Automation, Vol. Ra-2, No. 4, pp. 188-195.

Grupen, Roderic A. and Richard S. Weiss, 1991, "Force Domain Models for Multifingered Grasp Control", IEEE CH2969-4/91/0000/0418, pp. 418-423.

Holzmann, W. and J. Michael McCarthy, 1985, "Computing the Friction Forces Associated with a Three-Fingered Grasp", IEEE 1985 Journal of Robotics and Automation, Vol. Ra-1, No. 4, p 206-214.

Howe, Robert D., Imin Kao and Mark R. Cutkosky, 1988, "The Sliding of Robot Fingers Under Combined Torsion and Shear Loading", IEEE CH2555-1/88/0000/0103, pp. 103-105.

Jacobsen, S. C., J. E. Wood, D. F. Knutti and K. B. Biggers, 1984, "The UTAH/M.I.T. Dexterous Hands: Work in Progress", The International Journal of Robotics Research, Vol. 3, No. 4, pp. 21-50.

Kahng, Jay, and F. M. L. Amirouche, 1987, "Impact Force Analysis in Mechanical Hand Design", IEEE CH2413-3/87/0000/2061, pp. 2061-2067.

Kerr, D. R. and D. J. Sanger, 1987, " Restraint Analysis of a Rigid Body Using Frictional Elastic Contacts", ASME Journal of Mechanisms,Transmissions,and Automations in Design, Vol. 109, pp 450-454.

Kumar, Vijay and Kenneth J. Waldron, 1987, "Sub-Optimal Algorithms for Force Distribution in Multifingered Grippers", IEEE CH2413-3/87/0000/0252, pp. 252-257.

Mills, James K. and Andrew A. Goldenberg, 1989, "Force and Position Control of Manipulators During Constrained Motion Tasks", IEEE Transactions on Robotics and Automation, Vol. 5, No. 1, pp. 30-46.

Nakamura, Yoshihikoko, Kiyoshi Nagai and Tsuneo Yoshikawa, 1989, "Dynamics and Stability in Coordination of Multiple Robotic Mechanisms", The International Journal of Robotics Research, Vol. 8, No. 2, pp. 44-61.

Nguyen, Van-Duc, 1987, "Constructing Stable Grasps in 3D", IEEE CH2413-3/87/0000/0234, pp. 234-239.

Nguyen, Van-Duc, 1988, "Constructing Force-Closure Grasps", The International Journal of Robotics Research, Vol. 7, No.3, pp. 3-16.

Paul, Richard P., 1981, Robot Manipulators : Mathematics, Programming, And Control, The MIT Press, Cambridge, MA.

Payandeh, Shahram and Andrew A. Goldenberg, 1991, "A robust Force Controller : Theory and Experiments", IEEE CH2969-4/91/0000/0036, pp. 36-41.

Raibert, M. H. and J. J. Craig, 1981, "Hybrid Position/Force Control of Manipulators", ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 102, pp. 127-133.

Romdhane, L., 1989, "Kinestatic Analysis of Multi-Fingered Hands", Ph.D. Dissertation, University of Florida.

Sadamoto Kuni, 1984, Mechanical Hands Illustrated, Hemisphere Publising Corporation, New York.

Sinha, Pramath Raj and Jacob M. Abel, 1989, "A Contact Stress Model for Multifingered Grasps of Rough Objects", IEEE Transactions on Robotics and Automation Vol. 8, pp 7-22.

Stansfield, S. A., 1991, "Robotic Grasping of Unknown Objects: A Knowledge-Base Approach", The Internal Journal of Robotics Research , Vol. 10, No. 4, pp. 314-326.

Strang, Gilbert, 1980, Linear Algebra and Its Applications, Academic Press Inc., New York.

Vanderplaats, Garret N., 1984, Numerical Optimization Techniques For Engineering Design, Mcgraw-Hill, Inc., New York.

Yim, Wooson and Ali A. Seireg, 1988, "Tracking Moving Objects in 3-D Space", Computers in Mechanical Engineering, Vol. 7, No. 2, pp. 279-288.

Yoshikawa, Tsuneo and Kiyoshi Nagai, 1991, "Manipulating and Grasping Forces in Manipulation by Multifingered Robot Hands", IEEE Transactions on Robotics and Automation, Vol. 7, No. 1, pp. 67-77.

Yuan, M. S. C. and F. Freudenstein, 1971, "Kinematic Analysis of Spatial Mechanisms by Means of Screw Coordinates. Part 1 - Screw Coordinates", ASME Journal of Engineering for Industry, Vol. 93, pp. 61-66.

BIOGRAPHICAL SKETCH

Yun-Gyu Kim was born on Jan. 18, 1953, in Taegu, South Korea. He graduated in 1974 from SungKwang High School in Taegu, South Korea. After graduating from The 3rd Military Academy in 1976, he joined the Republic of Korea Army. As an instructor of The 3rd Military Academy, he earned his B.S. degree in 1982, and Master's degree in 1984 at the Kyungpook National University in Taegu. In 1989, he was awarded an Army scholarship to pursue the Ph.D. degree in the United States for three years. Currently he is a Ph.D. student in the Mechanical Engineering Department of the University of Florida.

He will go back to Korea after finishing his study and will continue his job as an assistant professor in The 3rd Military Academy.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Ali A. Seireg, Chair
Ebaugh Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph Duffy
Graduate Research Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Carl Crane
Associate Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Scott Smith
Associate Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Harold W Doddington

Harold W. Doddington
Engineer of Aerospace
Engineering, Mechanics,
and Engineering Science

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1993

Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School